



BEYOND AGGREGATE ACCURACY: EXPLAINABLE MINORITY-CLASS AND UNSEEN-SUBTYPE ANALYSIS IN NSL-KDD

Veljko Spasojević

Academy of Applied Studies Šumadija, Department Trstenik, Trstenik, Serbia
<https://orcid.org/0009-0004-6742-9546>

Milica Tufegdžić

Academy of Applied Studies Šumadija, Department Trstenik, Trstenik, Serbia
<https://orcid.org/0000-0003-3856-1498>

Marija Mojsilović

Academy of Applied Studies Šumadija, Department Trstenik, Trstenik, Serbia
<https://orcid.org/0009-0004-6818-2450>

Vladan Čolić

Academy of Applied Studies Šumadija, Department Trstenik, Trstenik, Serbia
<https://orcid.org/0009-0002-5125-1849>

ARTICLE INFO



Open access

JEL Category:
C52, C63, M15

Keywords:

artificial intelligence
data mining
tree algorithms
sensitivity analysis
network
information systems

ABSTRACT

Aggregate accuracy can conceal severe class-specific failures in machine-learning intrusion detection, particularly for rare or previously unseen attack patterns. Treating NSL-KDD as a diagnostic benchmark, this study re-evaluates Random Forest, XGBoost, and Decision Tree using leakage-free five-fold cross-validation on KDDTrain+ and final evaluation on untouched KDDTest+. XGBoost had the highest point estimates for accuracy (77.65%), weighted F1 (73.71%), and attack-versus-normal receiver operating characteristic area under the curve (ROC-AUC = 0.9678), while Decision Tree had the highest macro-F1 point estimate (56.46%). XGBoost nevertheless recalled only 6.83% of Remote-to-Local (R2L) attacks despite 98.01% precision. Within-family macro-class accuracy fell from 99.32% to 32.98% for seen versus unseen Denial-of-Service (DoS) subtypes and from 100% to 36.96% for Probe. An untuned weighting sensitivity analysis increased XGBoost macro-F1 to 59.49% and R2L F1 to 22.41%. Error-conditioned SHapley Additive exPlanations (SHAP) showed distinct attribution patterns between detected and missed R2L cases, supporting class-sensitive, generalization-aware benchmark evaluation. These findings are relevant to researchers and security practitioners interpreting aggregate intrusion-detection benchmark scores.

Address of the author:
Veljko Spasojević
vspasojevic@asss.edu.rs

Received: 19.08.2026
Revised: 25.08.2026
Accepted: 18.09.2026
Available online: 21.09.2026



1 INTRODUCTION

Machine-learning-based network intrusion detection systems (NIDSs) are often evaluated using aggregate measures such as accuracy, weighted F1 score, or the area under the binary attack-versus-normal receiver operating characteristic (ROC-AUC). These measures are useful, but they can give an incomplete picture when attack classes are highly imbalanced or when test traffic contains attack subtypes that were not represented during training. This matters operationally because a detector may achieve respectable aggregate performance while still missing the attack categories for which reliable detection is most difficult. Reviews of machine learning for intrusion detection have repeatedly identified class imbalance, dataset quality, generalization, and interpretability as persistent challenges (Abdelmoumin et al., 2022; Moustafa et al., 2023; Yang et al., 2022).

NSL-KDD remains widely used for controlled comparison even though it is an old benchmark derived from the KDD Cup 1999/Defense Advanced Research Projects Agency (DARPA) evaluation ecosystem. Its continued value is not that it represents contemporary production traffic, but that it provides a fixed train-test split with pronounced class imbalance and a deliberate generalization challenge: the test partition contains attack subtypes that do not occur in the training partition (Tavallaee et al., 2009; University of New Brunswick, Canadian Institute for Cybersecurity, n.d.). Recent work has emphasized that evaluation across changing or previously unseen traffic distributions can expose weaknesses that are obscured by conventional within-distribution testing (Chua & Salam, 2023).

We therefore treat NSL-KDD here as a diagnostic benchmark, not as evidence that any of these models is ready for deployment. Random Forest (RF), XGBoost (XGB), and Decision Tree (DT) are evaluated under a leakage-free protocol: preprocessing and model selection stay confined to KDDTrain+. KDDTest+ is used first for the primary post-selection evaluation of the three selected models and is then reused only for explicitly secondary diagnostic analyses. Beyond the usual aggregate numbers, we report macro-F1, balanced accuracy, and per-class metrics,

since these expose failures that weighted averages tend to hide. We also split attack subtypes into seen and unseen categories and compare them within the same macro families, reducing differences caused by macro-family composition. A class-weighted XGB variant is examined as an untuned sensitivity analysis rather than as a separately optimized fourth model. Finally, SHapley Additive exPlanations (SHAP) are used not just to rank features by class, but to compare what the model attributes to Remote-to-Local (R2L) cases it gets right versus those it misses. These secondary analyses are interpreted descriptively and are not treated as independent confirmatory tests.

The study addresses three research questions.

RQ1 asks whether aggregate and class-sensitive metrics produce the same conclusions about RF, XGB, and DT on the untouched KDDTest+ partition.

RQ2 asks how macro-class detection accuracy changes between seen and unseen attack subtypes when comparisons are controlled within Denial-of-Service (DoS), Probe, R2L, and User-to-Root (U2R) families.

RQ3 asks whether class weighting and error-conditioned SHAP analysis can clarify the nature of minority-class failure.

The goal is not to propose a new intrusion detection architecture, but to provide a more defensible and informative evaluation of familiar tree-based models on a benchmark whose structure makes evaluation choices especially consequential. From an IT-management perspective, the framework treats NIDS selection as a risk-sensitive decision in which minority-class false negatives and generalization failures are considered alongside aggregate predictive performance.

2 RELATED WORK

Tree-based models remain a common baseline for network intrusion detection, and for good reason. RF combines bootstrap aggregation with random feature selection and is a strong tabular baseline (Breiman, 2001); XGBoost, built on regularized gradient-boosted trees, has become one of the more widely used high-performance methods for structured data (Chen & Guestrin, 2016). But

which model "wins" depends heavily on the dataset, class structure, and metric used — recent comparative NIDS studies keep confirming that there is no universally superior classifier (Al Lail et al., 2023). This matters especially on NSL-KDD, where Normal and DoS dominate the training data while R2L and U2R barely register. Earlier work on adaptive cyber-attack detection has likewise explored learning-based recognition of attack classes using feature representations (Petrenko, 2017).

Class imbalance is therefore not a secondary detail. Standard classifiers trained on skewed data tend to favor majority classes, motivating techniques such as resampling, synthetic oversampling, and cost-sensitive or class-weighted learning (Chawla et al., 2002; Feng et al., 2021). In intrusion detection specifically, recent work on large and imbalanced datasets has demonstrated that imbalance-aware preprocessing can materially change minority-class results (Talukder et al., 2024). More recently, Silva et al. (2026) evaluated nine automated machine learning (AutoML) frameworks under the original imbalanced five-class NSL-KDD setting and likewise showed that strong aggregate or weighted performance can coexist with weak discrimination of rare attack classes such as R2L and U2R. Their findings further reinforce the need for minority-sensitive evaluation, while the present study focuses on fixed-model failure analysis, seen-versus-unseen subtype generalization, and explanation of misclassified minority-class cases. At the same time, improvements on an imbalanced benchmark need to be interpreted cautiously: a method can improve average class sensitivity without solving the separate problem of generalizing to attack behaviors that are absent from training.

A recent multiclass NSL-KDD study by Jyoti et al. (2026) likewise reported persistent detection difficulties for R2L and U2R despite feature optimization and evaluation across multiple classifiers, further illustrating that minority-class weakness remains relevant even in more extensively optimized pipelines.

Generalization to unseen attacks is a different problem altogether from class imbalance. Chua and Salam (2023) found that models performing well within a dataset can fall apart once the

distribution shifts more aggressively. NSL-KDD has a smaller-scale version of this same challenge built in: several attack subtypes show up only in KDDTest+, never in training (Tavallaei et al., 2009). The obvious approach — just comparing seen versus unseen subtypes globally — can be misleading, though, if the two groups happen to differ in their DoS/Probe/R2L/U2R composition. That is why we compare seen and unseen subtypes within each macro family rather than pooling everything together.

A parallel research stream focuses on explainability. SHAP provides additive feature attributions grounded in Shapley values and supports both local and aggregated explanations (Lundberg & Lee, 2017), while Local Interpretable Model-Agnostic Explanations (LIME) offers local surrogate explanations for individual predictions (Ribeiro et al., 2016). These approaches have increasingly been incorporated into intrusion detection frameworks. Keshk et al. (2023) combined global and local explainability in an Internet of Things (IoT) intrusion detection framework validated on multiple datasets, including NSL-KDD. Arreche et al. (2024) proposed an explainable artificial intelligence (XAI) framework across several models and intrusion datasets, and Mohale and Obagbuwa (2025) evaluated multiple machine-learning models using SHAP, LIME, and ELI5. Hermosilla et al. (2025) further examined explanation consistency for XGBoost and TabNet.

Recent work also shows that neither class-specific explainability nor imbalance-aware learning is sufficient as a novelty claim by itself. Hu et al. (2026), for example, used attack-specific SHAP analysis together with class balancing in an XGBoost-based IoT NIDS. Misclassification-focused explainability has also been examined directly. Mia et al. (2024) used SHAP-based visual analysis to diagnose false-positive and false-negative intrusion-detection outcomes. The present error-conditioned analysis differs by comparing aggregated attributions within the same true minority class between correctly detected and missed cases under the same fitted model. Accordingly, the present study does not claim methodological novelty for SHAP, class weighting, or tree-based classification in isolation. Its contribution is integrative and diagnostic: a leakage-free train-test protocol is combined with

within-family seen-versus-unseen subtype analysis, a controlled class-weighting sensitivity test, and error-conditioned explanation of minority-class outcomes. The objective is to examine how conclusions about model reliability change when aggregate performance, class imbalance, subtype novelty, and failure explanations are considered together.

3 DATA AND EXPERIMENTAL DESIGN

3.1 Dataset and class taxonomy

The experiments use the standard KDDTrain+ and KDDTest+ partitions of NSL-KDD. KDDTrain+ contains 125,973 network connection records, and KDDTest+ contains 22,544 records. Each connection contains 41 predictor attributes, including 38 numeric attributes and three categorical attributes (protocol_type, service, and

flag), plus the attack label and a difficulty field. The difficulty field was excluded from modeling because it is benchmark metadata rather than an observable traffic predictor. NSL-KDD was introduced to reduce redundancy and other limitations of KDD Cup 1999 while preserving a challenging train-test split (Tavallaee et al., 2009).

Attack subtype labels were mapped explicitly to five macro classes: Normal, DoS, Probe, R2L, and U2R. All 22,544 KDDTest+ records were retained. The taxonomy included test-only subtypes such as apache2, mailbomb, mscan, saint, httptunnel, snmpgetattack, ps, sqlattack, and xterm rather than discarding them because they were absent from KDDTrain+. This preserves the benchmark's intended generalization challenge. The resulting class distributions are shown in Table 1. KDDTrain+ contains 22 attack subtypes, whereas KDDTest+ contains 37 attack subtypes.

Table 1. NSL-KDD class distribution used in the final experiment.

Class	Train N	Train %	Test N	Test %
Normal	67,343	53.46%	9,711	43.08%
DoS	45,927	36.46%	7,460	33.09%
Probe	11,656	9.25%	2,421	10.74%
R2L	995	0.79%	2,885	12.80%
U2R	52	0.04%	67	0.30%
Total	125,973	100.00%	22,544	100.00%

Source: Authors' calculations based on NSL-KDD (Tavallaee et al., 2009).

3.2 Leakage-free preprocessing and model selection

The three categorical features were encoded using one-hot encoding with unknown categories ignored at transformation time. Numeric features were standardized to zero mean and unit variance. Both operations were placed inside a scikit-learn ColumnTransformer and Pipeline. Consequently, category discovery and scaling were fitted separately within each training fold during cross-validation and were never fitted on KDDTest+. Although tree models do not require standardization to define split orderings, retaining a single fold-fitted preprocessing pipeline ensures a consistent and reproducible transformation across all three classifiers.

RF, XGB, and DT were selected as representative tree-based models with different complexity and

interpretability characteristics. RF combines many decorrelated trees (Breiman, 2001), XGB implements regularized gradient boosting (Chen & Guestrin, 2016), and DT provides a single directly inspectable tree. To avoid using KDDTest+ for model selection, three prespecified configurations were evaluated for each algorithm using stratified five-fold cross-validation (CV) on KDDTrain+ only. Mean macro-F1 was the primary selection criterion because it gives equal weight to the five classes and is more sensitive than accuracy to minority-class failure. The selected settings were RF with 200 trees, maximum depth 20, and minimum leaf size 1; XGB with 250 trees, maximum depth 6, learning rate 0.1, subsample 0.8, column subsample 0.8, and minimum child weight 1; and DT with maximum depth 20 and minimum leaf size 2. After selection, each pipeline was fitted once on the full KDDTrain+ partition and evaluated on untouched KDDTest+.

3.3 Evaluation metrics and secondary analyses

Final evaluation included accuracy, weighted F1, macro-F1, balanced accuracy, per-class precision, recall, and F1, and binary attack-versus-normal ROC-AUC. For the binary attack-versus-normal ROC-AUC, the five-class labels were collapsed only at the evaluation stage, with Normal treated as the negative class and DoS, Probe, R2L, and U2R jointly treated as the positive Attack class. The continuous attack score was calculated as one minus the model-predicted probability of the Normal class; no separate binary classifier was trained. Weighted F1 reflects the empirical test class distribution, whereas macro-F1 and balanced accuracy give equal weight to each class and therefore expose minority-class failures. To quantify uncertainty in the main KDDTest+ metrics, 95% percentile bootstrap confidence intervals were calculated from 2,000 case-resampling replicates of the fixed test predictions; models were not refitted during bootstrap. Pairwise exact McNemar tests were computed on the same predictions and accompanied by accuracy differences and Cohen's g for discordant pairs so that the magnitude of model differences is reported alongside p-values.

For generalization analysis, every test attack subtype was marked as seen if the same subtype appeared in KDDTrain+ and unseen otherwise. Among the 37 attack subtypes present in KDDTest+, 20 were also present in KDDTrain+ and were therefore classified as seen: back, buffer_overflow, ftp_write, guess_passwd, imap, ipsweep, land, loadmodule, multihop, neptune, nmap, perl, phf, pod, portsweep, rootkit, satan, smurf, teardrop, and warezmaster. The remaining 17 test subtypes were classified as unseen: apache2, httptunnel, mailbomb, mscan, named, processtable, ps, saint, sendmail, snmpgetattack, snmpguess, sqlattack, udpstorm, worm, xlock, xsnoop, and xterm. These explicit subtype lists are provided to make the seen-versus-unseen partition fully reproducible. A global seen-versus-unseen comparison was calculated as a descriptive check, but the primary analysis was performed within DoS, Probe, R2L, and U2R families. Within-family aggregation weights subtype accuracies by the number of test examples in each subtype. This design reduces

the risk of attributing a difference to subtype novelty when it is partly caused by different macro-class composition. Accuracy in this analysis denotes correct prediction of the corresponding macro class, not identification of the individual attack subtype.

A class-weighted XGBoost sensitivity analysis used the already selected XGB configuration with inverse-frequency balanced sample weights computed from KDDTrain+ as $w_c = \frac{N}{Kn_c}$, where w_c is the weight assigned to class c , N is the number of training observations, $K = 5$ is the number of classes, and n_c is the number of training observations in class c . The weighted variant was not separately retuned and is therefore treated as a sensitivity experiment rather than as a fourth primary model. This distinction is important because the purpose is to test whether a simple imbalance-aware intervention changes the observed failure pattern, not to maximize benchmark performance.

3.4 Explainability analysis

SHAP analysis was performed on the selected standard XGB pipeline using TreeExplainer (Lundberg & Lee, 2017). TreeExplainer was used with its default raw model output; accordingly, the reported SHAP values represent contributions to class-specific raw model outputs rather than predicted probabilities. A reproducible class-stratified sample was drawn from KDDTest+: 500 examples each for Normal, DoS, Probe, and R2L, plus all 67 U2R examples, for a total of 2,067 observations. The stratified design intentionally gives minority classes more influence than their natural test prevalence, so aggregate SHAP summaries are interpreted as class-stratified diagnostic summaries rather than as estimates of feature importance under the empirical KDDTest+ distribution. Because one-hot encoding expands categorical variables, SHAP values for transformed dummy columns were aggregated back to the original 41 NSL-KDD feature groups by summing signed contributions belonging to the same source feature before absolute aggregation. Specifically, for each observation and model output, signed SHAP values belonging to dummy variables derived from the same original categorical feature were summed first, after which absolute values were used for importance

aggregation. This prevents categorical features from receiving artificially inflated aggregate importance solely because one-hot encoding expands them into multiple columns. Three complementary views were generated: a class-stratified aggregate ranking, SHAP importance by model output class, and true-class-conditioned explanations. In addition, R2L and U2R examples were split into correctly classified and missed groups, and mean absolute SHAP values for the corresponding class output were compared. Thus, the error-conditioned analysis compares attribution patterns within the same true class and under the same fitted model, rather than comparing explanations across separately trained classifiers. Error-conditioned SHAP is used here as a diagnostic association: differences in attribution patterns are not interpreted as causal effects. For R2L only, stability of correct-versus-missed differences was assessed with 2,000 group-resampling bootstrap replicates, resampling the correct and missed groups separately and calculating 95% percentile intervals for the difference in mean absolute SHAP attribution (correct minus missed). Because only nine U2R cases were correctly classified in the stratified sample, U2R error-conditioned results remain descriptive and exploratory without bootstrap inference.

All experiments used `random_state = 42`. The final software environment consisted of Python 3.11.0, NumPy 2.4.4, pandas 3.0.3, scikit-learn 1.8.0, XGBoost 3.2.0, SHAP 0.51.0, SciPy 1.17.1, and Matplotlib 3.10.9 on Windows 10. The complete analysis code, including the script used to regenerate the publication figures and the

required software dependencies, is available from the corresponding author upon reasonable request.

4 RESULTS

4.1 Training-only selection and final test performance

The cross-validation stage produced mean $\pm$ SD macro-F1 values of 0.9157 ± 0.0410 for the selected RF configuration, 0.9368 ± 0.0264 for XGB, and 0.8986 ± 0.0376 for DT. The corresponding mean balanced accuracies were 0.8863, 0.9251, and 0.8842. These values were computed exclusively on KDDTrain+ folds. When the selected pipelines were transferred to the untouched KDDTest+ partition, performance decreased substantially, consistent with the known train-test distribution shift in NSL-KDD.

Table 2 reports the final test results. XGB had the highest accuracy (77.65%), weighted F1 (73.71%), and attack-versus-normal ROC-AUC (0.9678) point estimates. DT had the highest macro-F1 (56.46%) and balanced accuracy (53.57%) point estimates, although its binary ROC-AUC was much lower (0.7463). The bootstrap 95% intervals for XGB and DT overlapped on balanced accuracy (51.89–55.34% versus 51.69–55.67%) and macro-F1 (53.07–58.16% versus 53.71–59.06%), so the small DT advantage in those point estimates is not interpreted as definitive model superiority. RF had the lowest accuracy (74.86%) and macro-F1 (49.18%) point estimates among the three. Thus, no single metric produces a complete ranking.

Table 2. Training-only model selection and final KDDTest+ performance.

Model	CV macro-F1 (%)	Accuracy (%)	Balanced acc. (%)	Weighted F1 (%)	Macro-F1 (%)	Attack/Normal AUC
RF	91.57 ± 4.10	74.86	48.43	70.50	49.18	0.9656
XGB	93.68 ± 2.64	77.65	53.45	73.71	55.56	0.9678
DT	89.86 ± 3.76	76.55	53.57	72.38	56.46	0.7463

Source: Authors' analysis. CV macro-F1: mean $\pm$ SD on KDDTrain+; other estimates: KDDTest+. Bootstrap 95% CIs (2,000 resamples; accuracy / balanced accuracy / weighted F1 / macro-F1 / AUC): RF 74.26–75.42 / 47.61–49.51 / 69.82–71.13 / 47.80–50.97 / 0.9632–0.9678; XGB 77.10–78.21 / 51.89–55.34 / 73.05–74.36 / 53.07–58.16 / 0.9656–0.9700; DT 75.98–77.08 / 51.69–55.67 / 71.70–73.00 / 53.71–59.06 / 0.7392–0.7532.

Pairwise exact McNemar tests found differences in error patterns for all model pairs (all $p < .001$). Effect magnitudes were also reported: XGB

exceeded RF accuracy by 2.79 percentage points (Cohen's $g = 0.444$), DT exceeded RF by 1.69 points ($g = 0.269$), and XGB exceeded DT by 1.10

points ($g = 0.207$). These paired magnitudes are reported alongside significance because the large KDDTest+ sample makes p-values alone insufficient for judging practical separation.

4.2 Class-level performance and minority-class failure

Figure 1 and Table 3 show the class-level results. XGB produced F1 scores of 80.07% for Normal,

89.81% for DoS, and 72.64% for Probe, but only 12.77% for R2L and 22.50% for U2R. The R2L result is particularly revealing because precision was 98.01% while recall was only 6.83%. In other words, the model was highly precise when predicting R2L, but it correctly identified only 6.83% of true R2L connections. This is a different failure mode from low precision: the principal problem is missed detection rather than excessive false alarms.

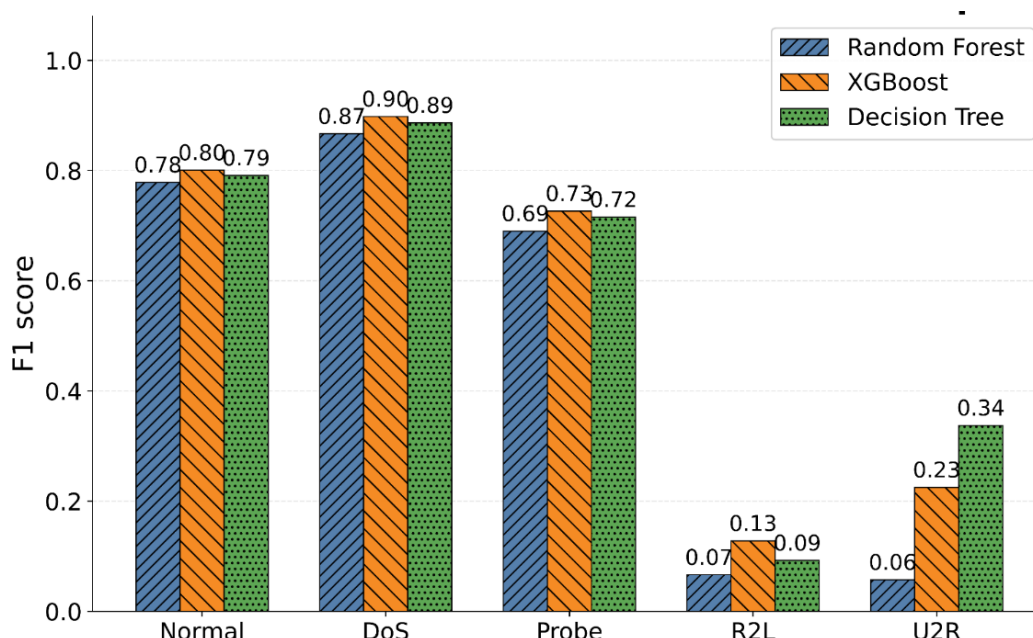


Figure 1. Per-class F1 scores for RF, XGBoost, and DT on KDDTest+.

Source: Authors' analysis.

Table 3. XGBoost class-level performance on KDDTest+.

Class	Precision (%)	Recall (%)	F1 (%)	Support
Normal	68.07	97.20	80.07	9,711
DoS	96.43	84.03	89.81	7,460
Probe	81.14	65.76	72.64	2,421
R2L	98.01	6.83	12.77	2,885
U2R	69.23	13.43	22.50	67

Source: Authors' analysis.

The other models exhibited the same broad pattern. RF achieved only 6.63% R2L F1 and 5.71% U2R F1. DT improved U2R F1 to 33.73% and therefore had a slightly higher macro-F1 point estimate than XGB, but its R2L F1 remained only 9.23%. The minority-class problem is therefore not eliminated by changing among these three tree-based architectures. Aggregate metrics are dominated by Normal and DoS, which together

account for more than three quarters of KDDTest+, while the operationally difficult R2L and U2R categories contribute relatively little to weighted averages. The confusion matrices in Figure 2 make the error structure explicit. For XGB, 2,683 of 2,885 R2L examples (93.0%) and 55 of 67 U2R examples (82.1%) were assigned to Normal. RF and DT similarly assigned 96.5% and 94.3% of R2L examples to Normal, respectively.

The dominant minority-class failure is therefore rather than diffuse confusion among attack false-negative assignment to the Normal class families.

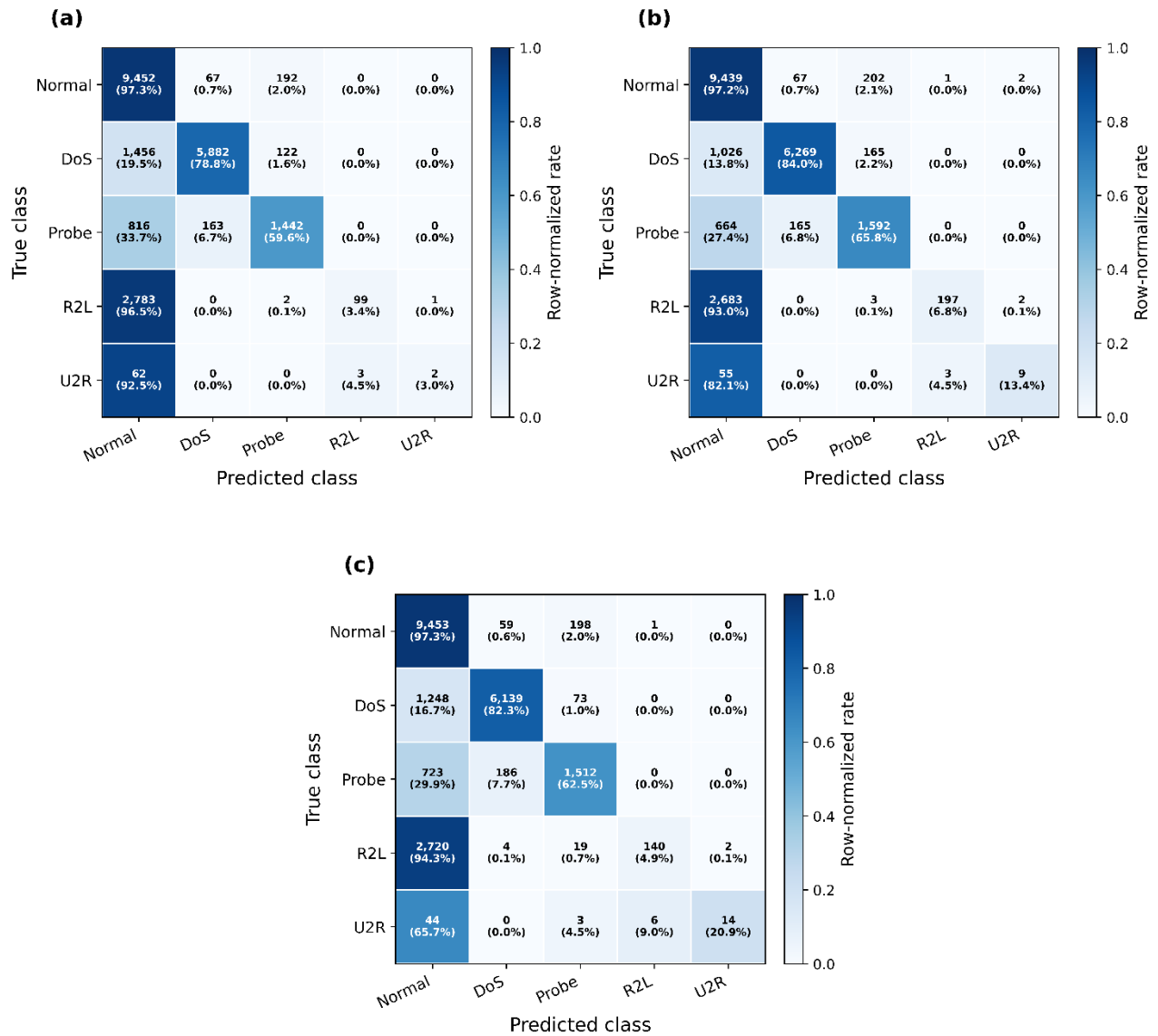


Figure 2. Row-normalized confusion matrices on KDDTest+: (a) Random Forest (RF), (b) XGBoost, and (c) Decision Tree (DT). Cells show counts and within-class percentages.

Source: Authors' analysis. Percentages are normalized within each true class (row).

4.3 Generalization to previously unseen attack subtypes

A naive aggregation across all attacks showed a large gap between subtypes seen and unseen during training, but the two groups contain different family compositions. Table 4 and Figure 3 therefore report the family-controlled XGB comparison. The generalization gap remained large. For DoS, XGB classified 99.32% of test

examples belonging to seen subtypes correctly but only 32.98% of examples from unseen subtypes. For Probe, the corresponding values were 100% and 36.96%. R2L was difficult even when the subtype had been seen during training (8.91%), and accuracy on unseen R2L subtypes fell to 0.15%. These comparisons are observational and do not isolate a causal effect of prior subtype exposure.

Table 4. XGBoost macro-class accuracy for records from seen and unseen subtypes within attack families.

Family	Seen N	Seen accuracy (%)	Unseen N	Unseen accuracy (%)
DoS	5,741	99.32	1,719	32.98
Probe	1,106	100.00	1,315	36.96
R2L	2,199	8.91	686	0.15
U2R	37	10.81	30	16.67

Source: Authors' analysis. U2R estimates are based on small samples and should be interpreted cautiously.

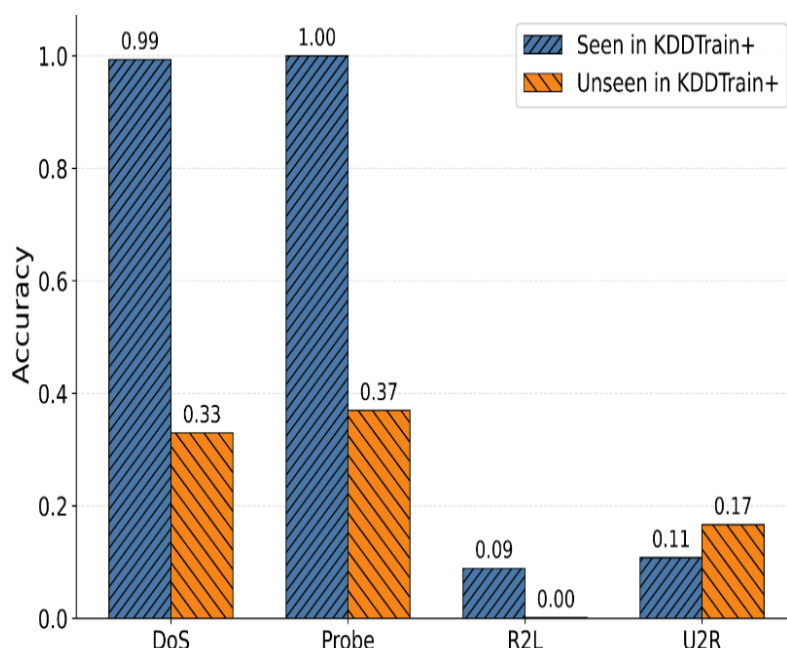


Figure 3. XGBoost macro-class accuracy for records from attack subtypes seen and unseen during training, compared within each attack family.

Source: Authors' analysis.

The subtype-level results also show that unseen does not mean uniformly undetectable. XGB correctly assigned 99.37% of records from the unseen saint subtype to Probe and 76.93% of records from the unseen apache2 subtype to DoS, while the corresponding rates were 16.97% for mscan and 0% for mailbomb and processtable. Most unseen R2L subtypes were also near zero. Generalization to unseen subtypes is therefore heterogeneous and subtype-dependent rather than a simple binary property. However, the seen-versus-unseen difference cannot be attributed solely to whether a subtype appeared in training: the groups also differ in sample size, intrinsic difficulty, feature distributions, and similarity to patterns learned from other subtypes. Seen status is therefore one possible contributor rather than an isolated causal explanation. U2R results are not interpreted strongly because the family contains

only 67 test examples in total, split into 37 seen-subtype and 30 unseen-subtype cases.

4.4 Class-weighted Sensitivity

XGBoost

Table 5 and Figure 4 compare the selected standard XGB with the same configuration trained using balanced inverse-frequency sample weights. In this untuned sensitivity configuration, the weighted variant yielded accuracy of 78.66% versus 77.65% for standard XGB, weighted F1 of 75.48% versus 73.71%, balanced accuracy of 56.43% versus 53.45%, and macro-F1 of 59.49% versus 55.56%. The changes were concentrated in minority-sensitive metrics rather than being driven solely by the majority classes; the comparison is not presented as a contest between separately optimized models.

Table 5. XGBoost class-weighting sensitivity on KDDTest+.

Variant	Accuracy (%)	Balanced acc. (%)	Weighted F1 (%)	Macro-F1 (%)
Standard XGB	77.65	53.45	73.71	55.56
Class-weighted XGB	78.66	56.43	75.48	59.49

Source: Authors' analysis.

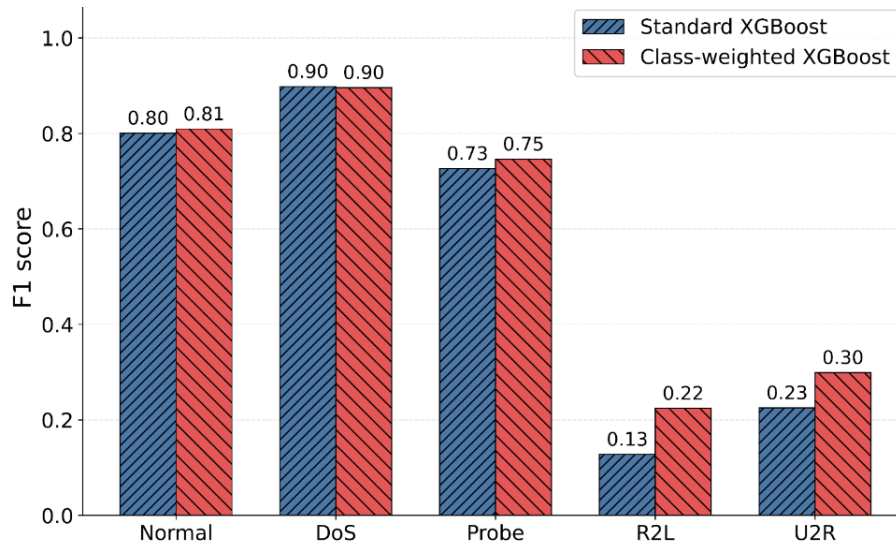


Figure 4. Per-class F1 sensitivity to balanced inverse-frequency sample weighting.

Source: Authors' analysis.

At the class level, the weighted sensitivity variant yielded R2L recall of 12.65% versus 6.83% and F1 of 22.41% versus 12.77% for standard XGB. U2R recall was 19.40% versus 13.43%, and F1 was 29.89% versus 22.50%. Probe F1 was 74.62% versus 72.64%, while DoS F1 was 89.59% versus 89.81%. The sensitivity result is consistent with weighting alleviating some minority-class failure in this fixed configuration, but because weighted XGB was not separately tuned, it is not interpreted as the best achievable weighted configuration and it does not make R2L or U2R detection reliable.

4.5 Class-specific and error-conditioned SHAP results

The class-stratified aggregate SHAP ranking identified service as the most influential original feature group (mean $|\text{SHAP}| = 2.1084$), followed by `src_bytes` (1.0255), `dst_host_same_src_port_rate` (0.4331), `count` (0.3661), `dst_host_error_rate` (0.3659), and `dst_host_srv_count` (0.3645). Because the SHAP sample intentionally contains approximately equal numbers of the four larger classes, these values

describe a diagnostic, class-stratified view rather than the natural KDDTest+ class mixture.

Figure 5 demonstrates that feature importance differs materially by XGB output class. This reinforces recent findings that attack-specific explanations can reveal model behavior that is hidden by a single global ranking (Hu et al., 2026). Because class-specific SHAP analysis is already established in IDS research, the following error-conditioned comparison is used here as a diagnostic extension rather than as a new XAI method: it contrasts attribution patterns for true minority-class examples according to whether the same fitted model classified them correctly or missed them.

For R2L, the SHAP sample contained 500 true R2L examples. Only 40 were classified correctly, and 460 were missed; all 460 missed R2L examples were predicted as Normal.

Figure 6 compares the R2L-output attributions for the two groups. Correctly detected R2L cases had larger mean absolute attribution for service (2.9291 versus 1.3182; correct-minus-missed $\Delta =$

1.6109, 95% bootstrap interval 1.3886–1.8274),
dst_host_same_src_port_rate (2.2568 versus
0.8316; $\Delta = 1.4252$, 1.2826–1.5420),
dst_host_srv_diff_host_rate (0.8724 versus
0.1253; $\Delta = 0.7471$, 0.6624–0.8243), and duration
(1.0548 versus 0.4059; $\Delta = 0.6489$, 0.4873–
0.7976). Conversely, dst_host_srv_count and

count had larger attribution among missed cases
($\Delta = -0.3487$, interval -0.3956 to -0.2973; and
 $\Delta = -0.2121$, interval -0.3244 to -0.0964,
respectively). These percentile intervals
summarize the stability of descriptive attribution
differences under group resampling and are not
causal tests.

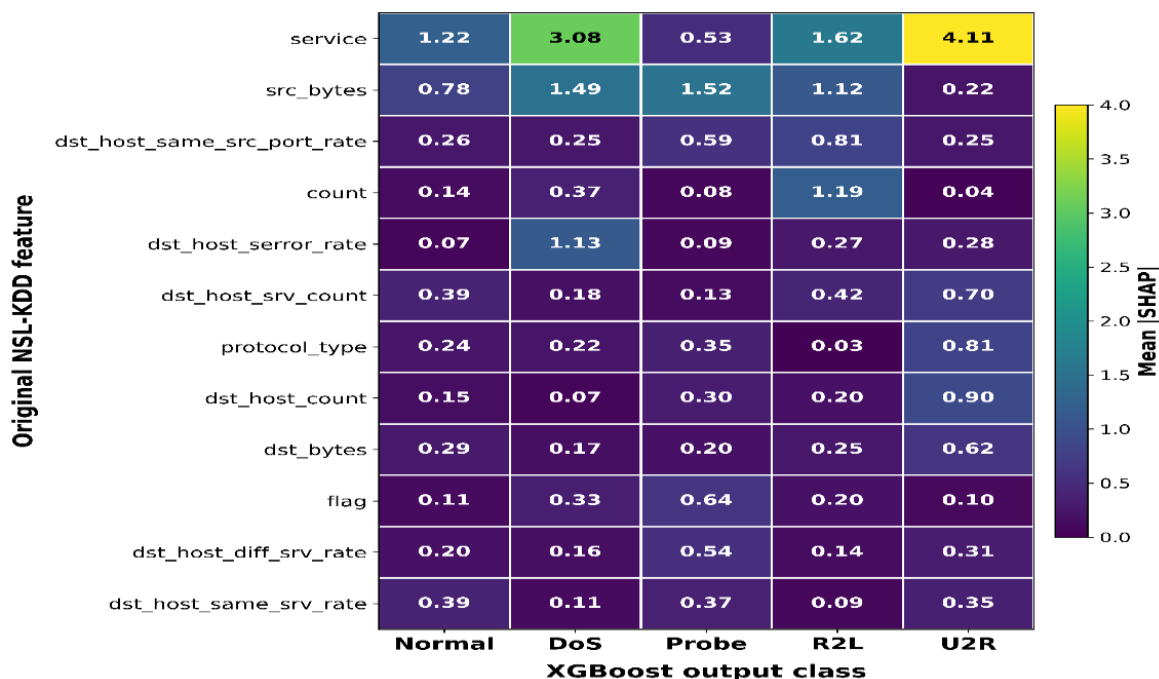


Figure 5. XGBoost class-specific SHAP importance for the twelve leading class-stratified feature groups

Source: Authors' analysis. The SHAP sample is class-stratified and is not weighted to the natural KDDTest+ class distribution.

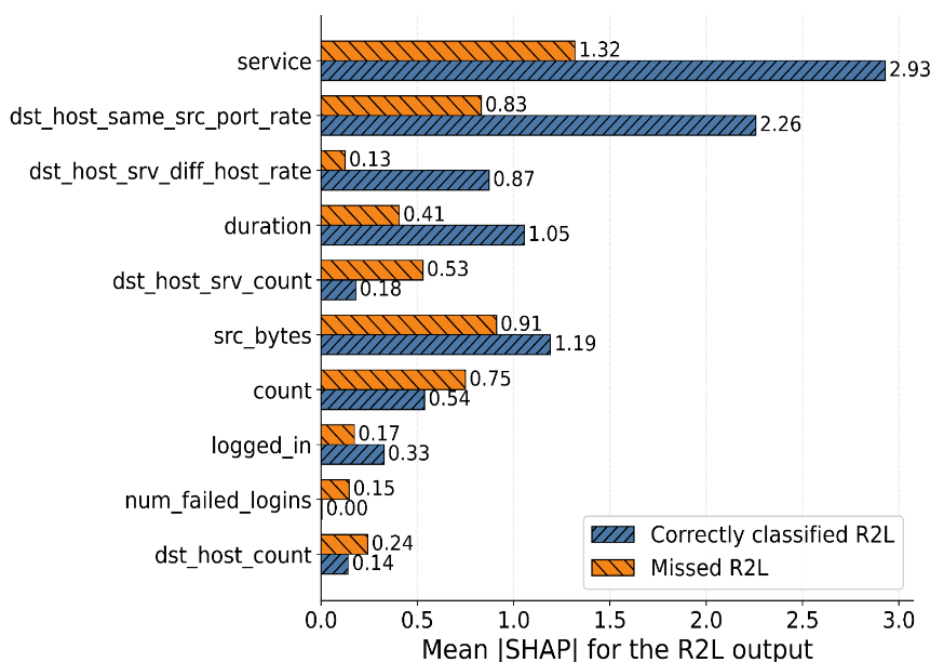


Figure 6. Error-conditioned R2L SHAP patterns for correctly classified and missed true R2L examples

Source: Authors' analysis. Mean absolute SHAP values refer to the R2L model output; Associations are descriptive, not causal.

The U2R error-conditioned analysis produced only nine correctly classified examples among 67 true U2R cases, so it is treated as exploratory. Fifty-five of the 58 missed U2R examples were classified as Normal. Features such as `root_shell`, `duration`, `num_compromised`, and `num_file_creations` showed stronger attribution in the small correct group, but the sample is too limited for strong inference. The R2L comparison is therefore the better-supported error-focused analysis in this study.

5 DISCUSSION

The results support the central premise that aggregate benchmark performance is not an adequate summary of intrusion detection behavior. XGB reached 77.65% accuracy and 73.71% weighted F1, figures that would appear reasonably competitive in a conventional model comparison. The same model, however, detected only 6.83% of R2L examples. The combination of 98.01% precision and 6.83% recall indicates a conservative R2L prediction pattern: the model predicts R2L rarely and misses most true R2L cases. Reporting only accuracy or weighted F1 would obscure this failure.

The difference between cross-validation and untouched test performance is also informative. Training-only cross-validation selected XGB with a macro-F1 of 0.9368, whereas its KDDTest+ macro-F1 was 0.5556. This gap should not be interpreted as a defect in cross-validation itself. Rather, it reflects that KDDTest+ is not exchangeable with KDDTrain+: class proportions differ substantially, and multiple attack subtypes are absent from training. The benchmark therefore illustrates a broader lesson emphasized by progressive-dataset evaluations: within-training validation can estimate generalization to similar samples but cannot guarantee resilience to future or structurally different attacks (Chua & Salam, 2023). More recently, Hung (2026) combined explainable tree-based intrusion detection with leave-one-attack-family-out evaluation on CICIoT2023. Unlike that binary family-holdout design, the present study retains the five-class NSL-KDD task and compares seen and unseen subtypes within the same macro families.

The family-controlled seen/unseen analysis sharpens this point. A global seen-versus-unseen

result could exaggerate or suppress the gap because unseen attacks are not distributed across families in the same way as seen attacks. After controlling the comparison within attack families, the difference remained pronounced for DoS and Probe. Yet some unseen subtypes, particularly `saint` and `apache2`, were recognized well. This heterogeneity argues against describing all test-only subtypes as equivalent 'zero-day' attacks. They are better understood as previously unseen benchmark subtypes whose observed detectability may reflect multiple factors, including feature similarity to learned patterns, sample size, intrinsic subtype difficulty, and prior subtype exposure.

Class weighting provides a complementary result. Imbalance-aware methods are well established (Chawla et al., 2002; Feng et al., 2021), and recent NIDS research continues to show benefits from rebalancing strategies (Talukder et al., 2024). In the present sensitivity configuration, the untuned weighted variant yielded R2L F1 of 22.41% versus 12.77% for standard XGB, while overall accuracy was 78.66% versus 77.65%. This pattern is consistent with class imbalance contributing to the observed minority-class failure, although the remaining R2L and U2R errors indicate that imbalance is not the sole explanation. Because the weighted variant was not separately tuned, these results characterize sensitivity to weighting rather than an optimally tuned comparison. The severe drop on unseen R2L subtypes and the large train-test shift indicate that representation and subtype coverage may also contribute to the observed failure pattern.

The SHAP results add a diagnostic layer to this interpretation. Prior XAI-IDS work has shown the value of global, local, and class-specific explanations for understanding high-performing models (Arreche et al., 2024; Keshk et al., 2023; Mohale & Obagbuwa, 2025), and recent XGBoost-based work has explicitly used class-specific SHAP for different attack types (Hu et al., 2026). The R2L error-conditioned analysis here does not claim a new explanation algorithm. Instead, it asks a narrower question: how do attributions differ between true R2L examples that the same model detects and those it misses? The four largest correct-minus-missed R2L contrasts had bootstrap stability intervals that remained away from zero, strengthening the descriptive stability of

those contrasts without changing their non-causal interpretation.

These SHAP differences are not causal evidence that modifying those features would turn a missed attack into a correct detection, nor do they prove that NSL-KDD lacks particular information. They reveal how the fitted XGB model distributes attribution under two observed outcome conditions. This distinction is important because post-hoc explanations can be informative without establishing causal mechanisms. The practical implication is that explainability can be used as an error-auditing tool: instead of asking only which features are globally important, analysts can inspect which attribution patterns are present when specific security-relevant classes are missed.

The findings also illustrate why conclusions about model superiority should be qualified. XGB had the highest point estimates for accuracy, weighted F1, and attack-versus-normal AUC, while DT had a marginally higher macro-F1 point estimate and balanced-accuracy point estimate. The XGB and DT bootstrap intervals for macro-F1 and balanced accuracy overlap, so neither model should be described as uniformly superior. DT also had a much lower binary AUC, and the U2R test support is very small ($n = 67$). Metric choice should therefore follow the operational objective. If missed rare attacks carry high cost, macro-F1, balanced accuracy, per-class recall, and error analysis deserve more weight than a single aggregate score.

5.1 Practical implications for security and IT management

Although the present results are benchmark-specific, they illustrate why operational NIDS selection should be treated as an IT risk-management decision rather than as a leaderboard exercise. Organizations should define class-specific false-negative tolerances, evaluate performance on rare and distribution-shifted traffic, and monitor whether model behavior changes as attack patterns evolve. From a governance perspective, per-class recall, balanced accuracy, and error-conditioned explanations provide relevant evidence: they help security teams identify where a model is reliable, where residual risk remains high, and when

additional controls, retraining, or human review are warranted.

6 LIMITATIONS AND THREATS TO VALIDITY

Several limitations constrain the interpretation of these results. First, NSL-KDD is derived from a historical benchmark ecosystem and does not represent contemporary encrypted traffic, modern application protocols, cloud workloads, or current adversarial behavior. The results should therefore be interpreted as a methodological benchmark analysis, not as an estimate of present-day deployment performance. Newer datasets are necessary for external validation (Al Lail et al., 2023; Moustafa et al., 2023).

Second, only three tree-based classifiers were examined. The design intentionally prioritizes a controlled comparison and interpretable error analysis rather than a broad search over deep architectures. Third, the hyperparameter search was deliberately small: three prespecified candidates per model were selected by training-only macro-F1. This prevents test-set tuning but does not guarantee globally optimal configurations. Fourth, the class-weighted XGB variant reused the selected standard-XGB hyperparameters and was not separately tuned; its results therefore demonstrate sensitivity to weighting rather than the best achievable weighted performance.

Fifth, the seen/unseen analysis is observational within a fixed benchmark. Family-controlled aggregation reduces macro-class composition confounding, but unseen subtypes still differ from seen subtypes in other ways, including sample size, intrinsic difficulty, and feature distributions. U2R is especially uncertain because only 67 test examples are available. Sixth, KDDTest+ is reused after the primary model evaluation for subtype, weighting, and SHAP diagnostics. These secondary analyses are therefore descriptive or hypothesis-generating rather than independent confirmatory tests, and their interpretation is deliberately limited. Finally, SHAP explanations describe the fitted model's attribution structure and are not causal. The R2L error-conditioned comparison is supported by 40 correct and 460 missed examples in the stratified sample and is supplemented by bootstrap stability intervals,

whereas the U2R comparison has only nine correct cases and should be considered exploratory.

7 CONCLUSION

This study re-evaluated RF, XGB, and DT on NSL-KDD using a train-only model-selection pipeline and an untouched 22,544-record KDDTest+ evaluation. XGB had the highest point estimates for accuracy (77.65%), weighted F1 (73.71%), and attack-versus-normal ROC-AUC (0.9678), whereas DT had slightly higher macro-F1 and balanced-accuracy point estimates. The corresponding XGB and DT bootstrap intervals for the two class-sensitive aggregate metrics overlapped, so these small differences do not establish uniform superiority. Severe minority-class failure remained: XGB R2L recall was only 6.83% despite 98.01% precision.

The analysis further showed that generalization to attack subtypes absent from training is heterogeneous but often poor even after controlling for attack family. XGB accuracy fell from 99.32% to 32.98% for seen versus unseen DoS subtypes and from 100% to 36.96% for Probe. These seen-versus-unseen differences are associations within a fixed benchmark and may also reflect sample size, feature-distribution differences, intrinsic subtype difficulty, and similarity to other learned patterns. The untuned

balanced-weight sensitivity variant yielded macro-F1 of 59.49%, and R2L F1 of 22.41%, a pattern consistent with class imbalance contributing to the failure without establishing that weighting is an optimally tuned solution.

Class-specific SHAP showed different attribution patterns across model outputs, while error-conditioned R2L analysis showed descriptive differences between successful and missed detections. Bootstrap stability intervals for the main R2L contrasts remained away from zero, but the SHAP findings remain non-causal; the U2R error-conditioned analysis remains exploratory because only nine cases were correctly classified. Within this diagnostic benchmark, the results support an evaluation principle that should be tested on newer datasets: aggregate performance should be complemented by class-sensitive metrics, explicit testing of generalization to unseen attack patterns, and explanation of failure cases. For operational adoption, NIDS selection should therefore be treated as an IT risk-management decision in which aggregate performance is only one component of evidence. Future work should validate the same analysis framework on contemporary intrusion datasets and investigate whether imbalance-aware training and richer temporal or session-level representations reduce the observed generalization gap.

Supplementary material

The supplementary material, including the source code and software requirements used in this study, is available from the authors upon reasonable request.

REFERENCES

- Abdelmoumin, G., Whitaker, J., Rawat, D. B., & Rahman, A. (2022). A survey on data-driven learning for intelligent network intrusion detection systems. *Electronics*, 11(2), Article 213. <https://doi.org/10.3390/electronics11020213>
- Al Lail, M., Garcia, A., & Olivo, S. (2023). Machine learning for network intrusion detection—A comparative study. *Future Internet*, 15(7), Article 243. <https://doi.org/10.3390/fi15070243>
- Arreche, O., Guntur, T., & Abdallah, M. (2024). XAI-IDS: Toward proposing an explainable artificial intelligence framework for enhancing network intrusion detection systems. *Applied Sciences*, 14(10), Article 4170. <https://doi.org/10.3390/app14104170>
- Breiman, L. (2001). Random forests. *Machine Learning*, 45(1), 5–32. <https://doi.org/10.1023/A:1010933404324>
- Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16, 321–357. <https://doi.org/10.1613/jair.953>

- 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* (pp. 785–794). Association for Computing Machinery. <https://doi.org/10.1145/2939672.2939785>
- Chua, T.-H., & Salam, I. (2023). Evaluation of machine learning algorithms in network-based intrusion detection using progressive dataset. *Symmetry*, 15(6), Article 1251. <https://doi.org/10.3390/sym15061251>
- Feng, Y., Zhou, M., & Tong, X. (2021). Imbalanced classification: A paradigm-based review. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 14(5), 383–406. <https://doi.org/10.1002/sam.11538>
- Hermosilla, P., Díaz, M., Berríos, S., & Allende-Cid, H. (2025). Use of explainable artificial intelligence for analyzing and explaining intrusion detection systems. *Computers*, 14(5), Article 160. <https://doi.org/10.3390/computers14050160>
- Hu, Y., Xiao, K., Luo, L., & Chen, L. (2026). An XGBoost-based intrusion detection framework with interpretability analysis for IoT networks. *Applied Sciences*, 16(2), Article 980. <https://doi.org/10.3390/app16020980>
- Hung, R.-J. (2026). An explainable XGBoost-based framework for IoT attack detection with unseen attack family evaluation. *Sensors*, 26(10), Article 3005. <https://doi.org/10.3390/s26103005>
- Jyoti, A., Singh, M., Banga, V. K., Kaur, J., Gowrishankar, J., Hota, S., Murugan, S., Bhowmik, A., & Ashok, N. (2026). Comparative performance evaluation of machine learning classifiers for multi-class intrusion detection on the NSL-KDD dataset. *Scientific Reports*. <https://doi.org/10.1038/s41598-026-59610-8>
- Keshk, M., Koroniotis, N., Pham, N., Moustafa, N., Turnbull, B. P., & Zomaya, A. Y. (2023). An explainable deep learning-enabled intrusion detection framework in IoT networks. *Information Sciences*, 639, Article 119000. <https://doi.org/10.1016/j.ins.2023.119000>
- Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. In I. Guyon, U. von Luxburg, S. Bengio, H. M. Wallach, R. Fergus, S. V. N. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems* 30 (pp. 4768–4777). Curran Associates, Inc. <https://doi.org/10.48550/arXiv.1705.07874>
- Mia, M., Pritom, M. M. A., Islam, T., & Hasan, K. (2024). Visually analyze SHAP plots to diagnose misclassifications in ML-based intrusion detection. In *2024 IEEE International Conference on Data Mining Workshops (ICDMW)* (pp. 632–641). IEEE. <https://doi.org/10.1109/ICDMW65004.2024.00088>
- Mohale, V. Z., & Obagbuwa, I. C. (2025). Evaluating machine learning-based intrusion detection systems with explainable AI: Enhancing transparency and interpretability. *Frontiers in Computer Science*, 7, Article 1520741. <https://doi.org/10.3389/fcomp.2025.1520741>
- Moustafa, N., Koroniotis, N., Keshk, M., Zomaya, A. Y., & Tari, Z. (2023). Explainable intrusion detection for cyber defences in the Internet of Things: Opportunities and solutions. *IEEE Communications Surveys & Tutorials*, 25(3), 1775–1807. <https://doi.org/10.1109/COMST.2023.3280465>
- Petrenko, T. (2017). Design of adaptive system for detection of cyber-attacks. *MEST Journal*, 5(1), 78–85. <https://doi.org/10.12709/mest.05.05.01.10>
- Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). “Why should I trust you?”: Explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 1135–1144). Association for Computing Machinery. <https://doi.org/10.1145/2939672.2939778>

- Silva, W. C., Vilas Boas, E. C., & de Figueiredo, F. A. P. (2026). *Evaluation of AutoML frameworks for IDS under imbalanced data conditions of the NSL-KDD dataset* [Preprint]. arXiv. <https://doi.org/10.48550/arXiv.2606.12611>
- Talukder, M. A., Islam, M. M., Uddin, M. A., Hasan, K. F., Sharmin, S., Alyami, S. A., & Moni, M. A. (2024). Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. *Journal of Big Data*, 11, Article 33. <https://doi.org/10.1186/s40537-024-00886-w>
- Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications* (pp. 1–6). IEEE. <https://doi.org/10.1109/CISDA.2009.5356528>
- University of New Brunswick, Canadian Institute for Cybersecurity. (n.d.). *ISCX NSL-KDD dataset 2009*. <https://www.unb.ca/cic/datasets/nsi.html>
- Yang, Z., Liu, X., Li, T., Wu, D., Wang, J., Zhao, Y., & Han, H. (2022). A systematic literature review of methods and datasets for anomaly-based network intrusion detection. *Computers & Security*, 116, 102675. <https://doi.org/10.1016/j.cose.2022.102675>