

Leakage-Aware Machine Learning for Imbalanced Network Intrusion Detection: A Comparative Study of Classical and Ensemble Classifiers

Pandhu Pramarta¹, Pujiastuti², Sri Mardiyati^{3*}

^{1,2,3} Department of Informatics Engineering, Faculty of Engineering and Computer Science, Universitas Indraprasta PGRI, South Jakarta City, Special Capital Region of Jakarta, Indonesia.

ABSTRACT

Network intrusion detection requires reliable machine learning models and evaluation procedures that provide overlap may affect the reliability of predictive evaluation. This study proposes a leakage-aware evaluation framework for binary network intrusion detection using the UNSW-NB15 dataset. Five model configurations were evaluated: Logistic Regression, Linear Support Vector Machine, Random Forest, and class-weighted variants of the two linear classifiers. Performance was assessed using the official test set and a leakage-controlled test set constructed by removing test records with exact predictor patterns found in the training data. The cross-split audit identified 1,302 overlapping feature patterns affecting 8,541 test records (10.37%). After their removal, 73,791 records remained for leakage-controlled evaluation. All five configurations showed lower reported accuracy, F1-score, and Macro F1-score under leakage-controlled evaluation. An ablation analysis further revealed different performance changes when label-consistent and label-conflicting overlapping records were removed. Random Forest achieved the strongest overall performance on the leakage-controlled test set, with 86.23% accuracy, an 88.11% F1-score, an 85.88% Macro F1-score, a 97.68% ROC-AUC, and a 97.87% average precision. Its attack recall was 98.37%, with a false negative rate of 1.63%; however, its false positive rate remained 26.85%, indicating a substantial potential burden from false alarms. Class weighting improved Macro F1 for both linear classifiers but reduced attack recall, demonstrating a trade-off between class-balanced performance and attack detection sensitivity. These findings highlight the importance of combining aggregate metrics with class-specific error measures and auditing exact feature-pattern overlap when evaluating intrusion detection models. Nevertheless, the framework was evaluated on a single benchmark dataset and does not establish generalizability across operational network environments. Future research should examine additional datasets, temporal data splits, multiclass settings, and previously unseen attack scenarios. Overall, the proposed framework provides a systematic approach to examining performance differences associated with class imbalance and exact cross-split feature overlap, supporting a more transparent interpretation of benchmark-based intrusion detection results

ARTICLE HISTORY

Received: 15 August 2026.

Revised: 17 August 2026.

Accepted: 26 September 2026.

Available Online: 20 October 2026.

Published: 1 December 2026

KEYWORDS

Class Imbalance; Data Leakage;
Network Intrusion Detection;
Random Forest; UNSW-NB15.

1. Introduction

The increasing frequency and sophistication of cyber threats have intensified the need for reliable Network Intrusion Detection Systems (NIDS) capable of identifying malicious activities in complex network environments. Traditional intrusion detection approaches that rely primarily on predefined rules and static signatures often have limited adaptability to evolving attack patterns, particularly previously unseen and rapidly changing threats. Consequently, machine learning and deep learning have increasingly been incorporated into NIDS to support automated pattern recognition and the detection of complex network attacks. Recent studies have demonstrated the effectiveness of machine learning, sequential deep learning, feature-selection-based models, and few-shot learning across different network environments (Hore *et al.*, 2024; Wijethilaka *et al.*, 2025; Xu *et al.*, 2025). These developments indicate the growing role of data-driven models in identifying heterogeneous and evolving network behavior. However, high predictive performance alone is insufficient to establish model reliability. The evaluation procedure must also provide a credible assessment of a model's ability to generalize to previously unseen network traffic. Two methodological challenges are particularly relevant to machine-learning-based intrusion detection: class imbalance and the reliability of train–test evaluation. Class imbalance may influence classifier behavior and reduce the ability to recognize underrepresented traffic patterns. Meanwhile, cross-split feature overlap, in which identical predictor patterns occur in both training and testing partitions, may affect the interpretation of test performance and produce overly optimistic estimates of generalization. These challenges require attention because a model's reported performance may depend not only on its classification capability but also on the composition and independence of the evaluation data. This study compares Logistic Regression, Linear Support Vector Machine (Linear SVM), Random Forest, and class-weighted variants of Logistic Regression and Linear SVM for binary network intrusion detection using the UNSW-NB15 dataset. The models are evaluated under both official and leakage-controlled testing conditions to examine performance differences associated with cross-split feature overlap. The study develops a leakage-aware experimental framework that audits exact overlap between training and testing predictor patterns, constructs a leakage-controlled test set, and quantifies changes in predictive performance after overlapping test records are removed. It also distinguishes the effects of label-consistent and label-conflicting overlap through ablation analysis and examines class-weighted learning as an approach to handling class imbalance. Model performance is assessed using accuracy, precision, recall, F1-score, Macro F1-score, and security-oriented metrics. By examining these factors within a unified experimental setting, the study aims to provide a more transparent assessment of classifier performance and generalization under imbalanced network traffic conditions.

Machine learning has become a prominent approach in Network Intrusion Detection Systems because it can learn discriminative patterns from network traffic and distinguish normal activities from malicious behavior. Classical classifiers such as Logistic Regression and Support Vector Machine, together with ensemble methods such as Random Forest, have been widely investigated because they offer different trade-offs in computational efficiency, predictive performance, and robustness. Ensemble-based approaches have demonstrated competitive intrusion detection performance by combining multiple decision structures to capture complex attack patterns (Ahmed *et al.*, 2024; Alsaffar *et al.*, 2024; Das *et al.*, 2022). Deep learning architectures incorporating Convolutional Neural Networks, Long Short-Term Memory, and Gated Recurrent Units have further expanded the range of approaches used to model complex network behavior (Sayem *et al.*, 2024). The UNSW-NB15 dataset has consequently become an important benchmark for evaluating intrusion detection models because it contains normal network traffic and multiple attack categories. Studies using this dataset have reported competitive performance from Random Forest and feature-oriented machine learning approaches (More *et al.*, 2024). Nevertheless, intrusion detection performance may vary across algorithms, datasets, and evaluation settings, emphasizing the importance of comparative assessment under controlled experimental conditions (Ahmad *et al.*, 2022; Layeghy *et al.*, 2023). Class imbalance remains a major challenge in machine-learning-based intrusion detection because unequal class distributions can bias classifier behavior and reduce its ability to recognize underrepresented traffic patterns. Various approaches, including resampling, ensemble methods, and class-weighted learning, have been proposed to address this problem (Le *et al.*, 2024; Li *et al.*, 2024; Rahma *et al.*, 2023). Class weighting adjusts the relative importance of classes during model training without generating synthetic observations or removing existing records. This approach provides an alternative to resampling when the objective is to retain the original training observations while modifying their influence on model optimization. Accordingly,

model evaluation should consider precision, recall, F1-score, and Macro F1-score alongside accuracy to provide a more informative assessment of classification performance across classes (Amien *et al.*, 2025). Data leakage is another concern in machine learning evaluation, particularly when identical or highly similar feature patterns occur across training and testing partitions. Such overlap may affect test-set independence and produce performance estimates that do not fully represent generalization to previously unseen patterns (Joeres *et al.*, 2025). In intrusion detection, leakage associated with data preparation or evaluation procedures may similarly compromise the reliability of reported classification results (Bouke & Abdullah, 2023). Auditing cross-split overlap is therefore relevant to assessing the extent to which evaluation results are associated with predictor patterns already represented in the training data. Examining the effects of overlap removal can also help distinguish performance changes associated with label-consistent patterns from those associated with label-conflicting patterns. Although previous studies have examined classifier performance, class imbalance, and data leakage in machine learning, the present study focuses on integrating these concerns within a single experimental framework for binary network intrusion detection. The framework combines exact train–test feature-overlap auditing, leakage-controlled evaluation, ablation analysis distinguishing label-consistent from label-conflicting overlap, and class-weighted learning for linear classifiers. This design enables a comparative examination of classifier performance under official and leakage-controlled testing conditions while assessing the distinct effects of overlap types and class weighting. Rather than treating aggregate predictive performance as the sole indicator of model reliability, the study also considers class-specific metrics and changes in performance after overlapping test records are removed. This integrated evaluation provides a basis for examining the limitations of benchmark-based performance estimates under class imbalance and cross-split feature overlap.

2. Methodology

2.1 Research Design and Dataset

This study employed a quantitative experimental design to evaluate machine learning classifiers for binary network intrusion detection under class imbalance and cross-split feature overlap. The publicly available UNSW-NB15 dataset was used as a secondary data source. The official partition was retained, comprising 175,341 training records and 82,332 test records. Each record initially contained 45 attributes representing network traffic characteristics and target information. The binary label variable served as the prediction target, with 0 representing Normal traffic and 1 representing Attack traffic. The id, attack_cat, and label variables were excluded from the predictor set, resulting in 42 predictors comprising 39 numerical variables and three categorical variables (proto, service, and state). The training set contained 56,000 Normal records and 119,341 Attack records, whereas the official test set contained 37,000 Normal records and 45,332 Attack records. After test observations associated with feature patterns overlapping with the training data were removed, the leakage-controlled test set comprised 73,791 records, including 35,528 Normal records and 38,263 Attack records. Table 1 summarizes the dataset composition and experimental characteristics used in the subsequent analyses.

Table 1. Dataset and Experimental Characteristics

Characteristic	Training Set	Official Test Set	Leakage-Controlled Test Set
Number of records	175,341	82,332	73,791
Initial attributes	45	45	45
Predictor variables	42	42	42
Numerical predictors	39	39	39
Categorical predictors	3	3	3
Normal records	56,000	37,000	35,528
Attack records	119,341	45,332	38,263

Source: UNSW-NB15 dataset and authors' processing (2026).

The training set exhibited a pronounced class imbalance, with Attack records substantially outnumbering Normal records. In contrast, the official test set had a more balanced class distribution. The leakage-control

procedure reduced the number of test observations while retaining records from both classes. Because differences in class composition and cross-split feature overlap may influence predictive performance, both test settings were retained to examine classifier performance under the original benchmark partition and a stricter evaluation setting based on non-overlapping feature patterns.

2.2 Leakage Audit and Data Preprocessing

Before model training, an exact cross-split feature-overlap audit was conducted between the official training and test sets using the 42 predictor variables. Each observation was represented by its predictor values, and feature vectors were hashed and compared across the two partitions to identify identical feature patterns. The comparison focused on predictor values rather than record identifiers or the target label. The audit identified 1,302 unique overlapping feature patterns affecting 8,541 test records, equivalent to 10.37% of the official test set. Overlapping test observations were excluded from the test partition to construct the leakage-controlled test set, while the original training set remained unchanged. The resulting test set contained 73,791 records whose exact predictor patterns were not identified in the training partition. This procedure was intended to assess model performance on feature patterns absent from the training data rather than to establish that every possible form of data leakage or dependence had been eliminated. Overlapping patterns were also distinguished according to their class labels. Label-consistent overlap referred to identical predictor patterns associated with the same class in the training and test partitions, whereas label-conflicting overlap referred to identical predictor patterns associated with different classes across the partitions. This distinction supported the subsequent ablation analysis, which examined the effects of removing label-consistent patterns, label-conflicting patterns, or all overlapping patterns on Random Forest performance. The predictor variables were separated from the target and excluded attributes before model development. The three categorical predictors (proto, service, and state) and 39 numerical predictors formed the 42-variable input representation. Preprocessing was performed using the training data, as specified in the research workflow, and the resulting representation was used for model evaluation.

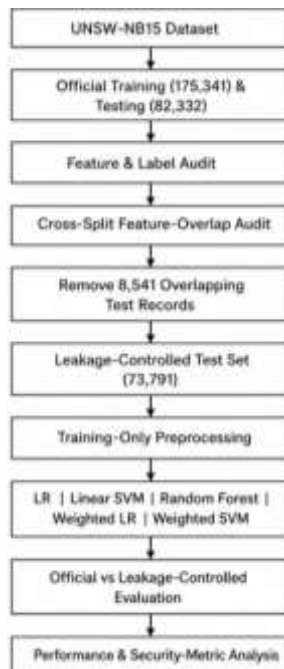


Figure 1. Research Framework

Figure 1 illustrates the research workflow, including the original dataset partition, feature and label auditing, cross-split feature-overlap identification, removal of overlapping test observations, training-based preprocessing, classifier training, and performance evaluation under the official and leakage-controlled test settings.

2.3 Classification Models and Class-Weighted Learning

Five model configurations were evaluated: Logistic Regression (LR), Linear Support Vector Machine (Linear SVM), Random Forest (RF), class-weighted Logistic Regression, and class-weighted Linear SVM. LR and Linear SVM represented linear classification approaches, whereas RF represented an ensemble-learning approach. The models were evaluated using the same predictor set and experimental data partitions to support a consistent comparison of their classification performance. The two class-weighted configurations were included to examine the effect of imbalance-aware learning without generating synthetic observations or removing existing training records. Balanced class weights were applied to LR and Linear SVM, while the corresponding unweighted configurations used no class-weight adjustment. The LR configurations used a maximum iteration limit of 1,000. The Linear SVM configurations used a regularization parameter of $(C=1.0)$ and a maximum iteration limit of 5,000. RF was configured with 200 decision trees and no additional class-weight treatment.

Table 2. Machine Learning Model Configuration

Model	Main Configuration	Imbalance Treatment
Logistic Regression	max_iter = 1,000	None
Class-Weighted LR	max_iter = 1,000	Balanced class weights
Linear SVM	$C = 1.0$; max_iter = 5,000	None
Class-Weighted Linear SVM	$C = 1.0$; max_iter = 5,000	Balanced class weights
Random Forest	200 decision trees	None

Source: Authors' experimental configuration (2026).

The comparison was designed to examine differences between linear and ensemble classification approaches and to assess whether class weighting changed the balance between class-specific performance measures. The reported parameter settings represent the configurations specified in the experimental description.

2.4 Model Evaluation and Reproducibility

Each trained model was evaluated on two test sets: the original official test set and the leakage-controlled test set. The same training partition and model configurations were retained across the two evaluation settings so that differences in the reported metrics could be examined in relation to the removal of overlapping test observations. Classification performance was assessed using accuracy, precision, recall, F1-score, and Macro F1-score. Differences between the official and leakage-controlled evaluation results were calculated to quantify changes in predictive performance following overlap control. The ablation analysis separately examined Random Forest performance under four scenarios: the complete official test set, removal of label-consistent overlapping records, removal of label-conflicting overlapping records, and removal of all overlapping records. Because missed attacks and false alarms have different implications for network intrusion detection, additional security-oriented metrics were calculated for the selected best-performing model. These metrics included specificity, false positive rate, false negative rate, negative predictive value, and balanced accuracy. ROC-AUC and average precision derived from the Precision–Recall analysis were also used to assess discrimination beyond the default classification threshold. Attack was treated as the positive class for the security-oriented evaluation. The five model configurations were compared using their reported classification metrics under the official and leakage-controlled test settings. Random Forest was identified as the best-performing configuration based on the comparative results reported in this study and was subsequently examined using class-specific metrics, a confusion matrix, and ROC and Precision–Recall curves. Its selection should be interpreted as a comparative finding within the evaluated configurations and dataset, rather than evidence of universal superiority over other intrusion detection approaches. The experiments were implemented in Python using scikit-learn and executed in Google Colab.

3. Result

The distinction between label-consistent and label-conflicting overlap is important for interpreting the potential influence of duplicated feature patterns on model evaluation. Label-consistent overlap may allow models to encounter previously observed predictor patterns associated with the same class, whereas label-conflicting overlap reflects cases in which identical predictor patterns are associated with different class labels across the two partitions. These two conditions may have different implications for the interpretation of classification performance. Therefore, separating the overlap types provides a more detailed basis for assessing evaluation reliability than considering feature overlap as a single category. The subsequent ablation analysis examines how excluding overlapping records affects model performance under the leakage-controlled evaluation setting.

3.1 Cross-Split Feature-Overlap Results

The cross-split feature-overlap audit identified duplicated predictor patterns between the official training and test partitions of the UNSW-NB15 dataset. Of the 82,332 observations in the official test set, 8,541 records (10.37%) contained feature patterns that were also present in the training set. After these records were excluded, 73,791 observations remained for leakage-controlled evaluation. At the pattern level, 1,302 unique feature patterns were shared across the two partitions. Of these, 1,023 patterns (78.57%) were label-consistent, meaning that identical predictor patterns were associated with the same class in both partitions. The remaining 279 patterns (21.43%) were label-conflicting, meaning that identical predictor patterns were associated with different classes. At the observation level, label-consistent patterns affected 8,015 test records (9.73% of the official test set), whereas label-conflicting patterns affected 526 records (0.64%). These results indicate that most observed overlap involved patterns with consistent class labels. The potential effects of these two overlap types were examined further through the ablation analysis.

Table 3. Cross-Split Feature-Overlap Results

Measurement	Result
Training records	175,341
Official test records	82,332
Unique overlapping patterns	1,302
Affected test records	8,541 (10.37%)
Label-consistent patterns	1,023 (78.57%)
Label-conflicting patterns	279 (21.43%)
Label-consistent affected records	8,015 (9.73%)
Label-conflicting affected records	526 (0.64%)
Leakage-controlled test records	73,791

Source: Authors' experimental results (2026).

3.2 Ablation Analysis of Cross-Split Feature Overlap

An ablation analysis was conducted to examine the effects of label-consistent and label-conflicting overlap on predictive performance. Random Forest was selected for this analysis because it achieved the strongest overall performance in the main classifier comparison. Four evaluation scenarios were considered: the complete official test set, removal of label-consistent overlapping records only, removal of label-conflicting overlapping records only, and removal of all overlapping records. The training data and model configuration were kept unchanged across the scenarios.

Table 4. Random Forest Ablation Analysis of Cross-Split Feature Overlap

Evaluation Scenario	N	Accuracy	F1-score	Macro F1	Δ Accuracy	Δ F1	Δ Macro F1
Official test	82,332	0.8713	0.8941	0.8651	0.0000	0.0000	0.0000
Remove consistent overlap	74,317	0.8575	0.8767	0.8539	-0.0139	-0.0174	-0.0112
Remove conflicting	81,806	0.8764	0.8983	0.8703	+0.0050	+0.0043	+0.0052

overlap							
Remove all overlap	73,791	0.8629	0.8816	0.8594	-0.0084	-0.0125	-0.0057

Source: Authors' experimental results (2026).

As shown in Table 4, removing label-consistent overlapping records was associated with the largest reduction in the reported performance metrics. Accuracy decreased from 0.8713 to 0.8575, F1-score from 0.8941 to 0.8767, and Macro F1 from 0.8651 to 0.8539. The corresponding reported changes were -0.0139, -0.0174, and -0.0112, respectively. These results suggest that the label-consistent patterns contributed to higher performance estimates under the official test-set evaluation. In contrast, removing label-conflicting overlapping records was associated with higher reported performance: accuracy increased to 0.8764, F1-score to 0.8983, and Macro F1 to 0.8703. This pattern is consistent with the possibility that label-conflicting observations introduced classification ambiguity. When all overlapping records were removed, the reported accuracy, F1-score, and Macro F1 decreased to 0.8629, 0.8816, and 0.8594, respectively.

3.3 Classification Performance

Five model configurations were evaluated using both the official and leakage-controlled test sets. Table 5 presents the comparative results for Logistic Regression, class-weighted Logistic Regression, Linear SVM, class-weighted Linear SVM, and Random Forest.

Table 5. Classification Performance on Official and Leakage-Controlled Test Sets

Model		Test Set	Accuracy	Precision	Recall	F1-score	Macro F1
Logistic Regression		Official	0.8097	0.7535	0.9726	0.8491	0.8012
Logistic Regression		Leakage-controlled	0.7946	0.7267	0.9677	0.8301	0.7906
Class-weighted LR		Official	0.8355	0.8033	0.9256	0.8601	0.8306
Class-weighted LR		Leakage-controlled	0.8232	0.7796	0.9189	0.8435	0.8215
Linear SVM		Official	0.8124	0.7510	0.9858	0.8525	0.8007
Linear SVM		Leakage-controlled	0.7975	0.7247	0.9828	0.8343	0.7904
Class-weighted SVM	Linear	Official	0.8300	0.7939	0.9286	0.8560	0.8266
Class-weighted SVM	Linear	Leakage-controlled	0.8171	0.7707	0.9214	0.8394	0.8171
Random Forest		Official	0.8708	0.8172	0.9859	0.8936	0.8645
Random Forest		Leakage-controlled	0.8623	0.7978	0.9837	0.8811	0.8588

Source: Authors' experimental results (2026).

Random Forest achieved the highest reported accuracy, F1-score, and Macro F1 under both evaluation settings. On the leakage-controlled test set, it recorded an accuracy of 0.8623, recall of 0.9837, F1-score of 0.8811, and Macro F1 of 0.8588.

3.4 Performance Changes After Leakage Control

Table 6 presents the differences between the official and leakage-controlled evaluation results. Negative values indicate lower reported performance after overlapping test records were excluded.

Table 6. Performance Changes After Leakage Control

Model	Δ Accuracy	Δ F1-score	Δ Macro F1
Logistic Regression	-0.0151	-0.0190	-0.0106
Class-weighted LR	-0.0123	-0.0182	-0.0091
Linear SVM	-0.0149	-0.0183	-0.0103
Class-weighted Linear SVM	-0.0129	-0.0187	-0.0095

Random Forest	-0.0085	-0.0126	-0.0057
---------------	---------	---------	---------

Source: Authors' experimental results (2026).

All five configurations exhibited lower reported accuracy, F1-score, and Macro F1 under leakage-controlled evaluation. Random Forest showed the smallest reported reductions, with changes of -0.0085 in accuracy, -0.0126 in F1-score, and -0.0057 in Macro F1. Logistic Regression showed the largest reported reductions in accuracy (-0.0151) and F1-score (-0.0190), while the other linear configurations exhibited intermediate changes. These results suggest that Random Forest was comparatively less affected by the removal of overlapping test records within the evaluated configurations. However, this finding does not establish immunity to data leakage. The results instead indicate that classifier performance differed between the official and leakage-controlled evaluation settings. Figure 2 presents the reported Macro F1-scores for the five configurations under both settings.

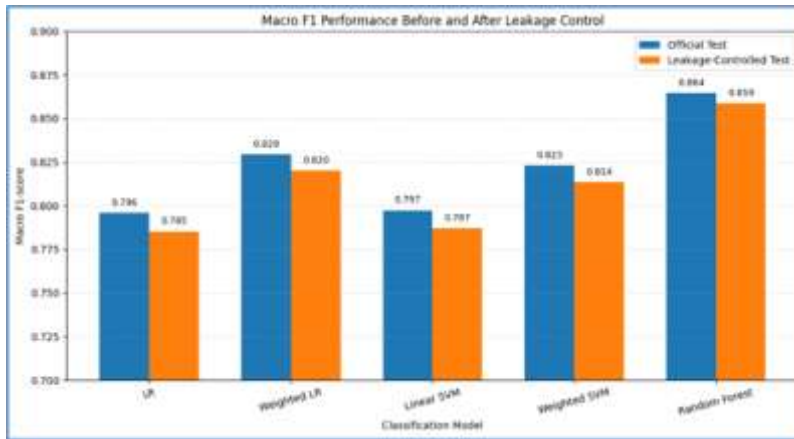


Figure 2. Macro F1 Performance Before and After Leakage Control

Source: Authors' experimental results (2026).

Random Forest recorded a Macro F1 of 0.8645 on the official test set and 0.8588 on the leakage-controlled test set. The same direction of change was reported for all five configurations. The numerical values in Figure 2 should match those in Table 5, while the changes should be consistent with Table 6 after accounting for rounding.

3.5 Best-Performing Model Evaluation

Random Forest achieved the highest reported overall performance among the evaluated configurations on the leakage-controlled test set and was therefore selected for further class-level evaluation. Figure 3 presents its confusion matrix, which reports 25,988 true negatives, 9,540 false positives, 623 false negatives, and 37,640 true positives.

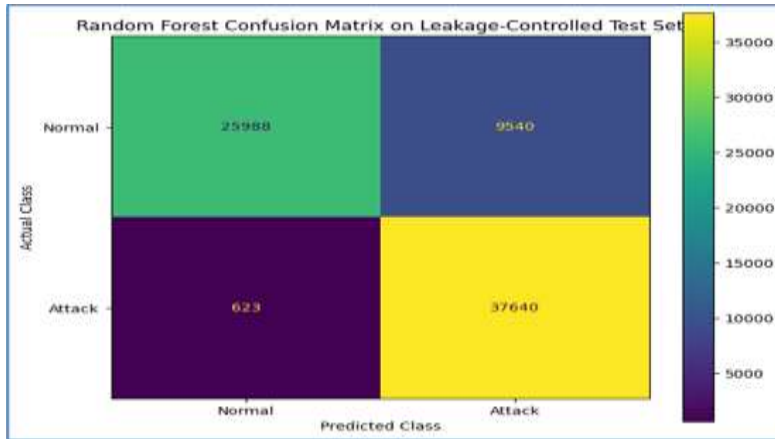


Figure 3. Random Forest Confusion Matrix on the Leakage-Controlled Test Set
Source: Authors' experimental results (2026).

The confusion matrix indicates that Random Forest correctly classified 37,640 Attack records and misclassified 623 Attack records as Normal. At the same time, 9,540 Normal records were incorrectly classified as Attack. This pattern indicates high sensitivity to Attack traffic alongside a substantial false-positive burden. In operational intrusion detection, false negatives represent attacks that remain undetected, whereas false positives can increase the volume of alerts requiring investigation. Consequently, the model's performance should be interpreted using both aggregate classification metrics and class-specific error measures. Table 7 presents the reported security-oriented metrics for Random Forest on the leakage-controlled test set.

Table 7. Security-Oriented Performance of Random Forest

Metric	Value
Accuracy	0.8623
Precision (Attack)	0.7978
Recall / Detection Rate	0.9837
F1-score	0.8811
Specificity	0.7315
False Positive Rate	0.2685
False Negative Rate	0.0163
Negative Predictive Value	0.9766
Balanced Accuracy	0.8576
ROC-AUC	0.9768
PR-AUC / Average Precision	0.9787

Source: Authors' experimental results (2026).

The reported ROC-AUC was 0.9768, while the reported average precision from the Precision–Recall analysis was 0.9787. Figure 4 shows the ROC curve, and Figure 5 shows the Precision–Recall curve for Random Forest on the leakage-controlled test set. These curves complement the threshold-dependent metrics by illustrating the model's discrimination performance across classification thresholds.

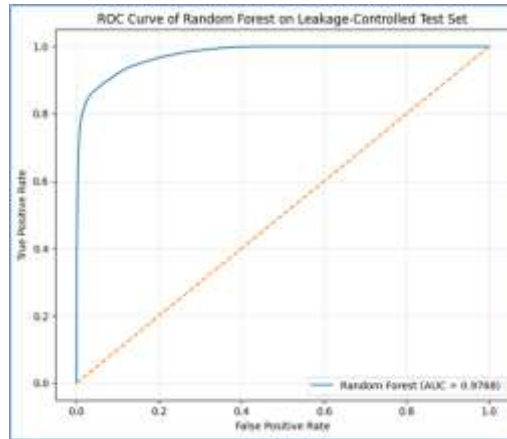


Figure 4. ROC Curve of Random Forest on the Leakage-Controlled Test Set
Source: Authors' experimental results (2026).



Figure 5. Precision–Recall Curve of Random Forest on the Leakage-Controlled Test Set
Source: Authors' experimental results (2026).

4. Discussion

4.1 Effect of Cross-Split Feature Overlap

The cross-split audit identified 1,302 exact feature patterns shared between the training and test partitions, affecting 8,541 test records (10.37% of the official test set). Of these overlapping patterns, 21.43% were label-conflicting, indicating that identical predictor patterns were associated with different class labels across the two partitions. These findings demonstrate that the official test partition contained feature patterns already represented in the training data and therefore did not exclusively represent previously unseen patterns. This observation is consistent with concerns that train–test contamination can compromise the reliability of machine learning generalization estimates when identical or related observations occur across data partitions (Joeres *et al.*, 2025; Bouke & Abdullah, 2023). The comparative evaluation showed lower reported performance for all five classifier configurations after overlapping test records were excluded. Logistic Regression exhibited reductions of 0.0151 in accuracy and 0.0190 in F1-score, whereas Random Forest showed reductions of 0.0085 and 0.0126, respectively. Although the magnitude of the changes differed across models, the consistent direction of the reported differences indicates that evaluation results depended on the composition of the test set. Therefore, reporting performance under both official and leakage-controlled evaluation settings provides a more informative assessment of classifier behavior on feature patterns not observed during training. The ablation analysis further indicated that the effects of

overlap differed according to label consistency. Removing label-consistent overlapping records was associated with lower reported performance, whereas removing label-conflicting records was associated with modest improvements. These results suggest that the two overlap types affected the evaluation differently. However, the findings should be interpreted as evidence of sensitivity to test-set composition rather than proof that all observed performance differences were caused exclusively by data leakage.

4.2 Effect of Class-Weighted Learning

Class-weighted learning improved Macro F1 for both linear classifiers, but this improvement was accompanied by lower Attack Recall. On the leakage-controlled test set, the Macro F1-score of Logistic Regression increased from 0.7906 to 0.8215 after class weighting, while that of Linear SVM increased from 0.7904 to 0.8171. However, Attack Recall decreased from 0.9677 to 0.9189 for Logistic Regression and from 0.9828 to 0.9214 for Linear SVM. These results indicate that improvements in Macro F1 did not correspond to uniform improvements in attack detection. Instead, class weighting changed the balance between aggregate class performance and sensitivity to Attack observations. This trade-off is important in network intrusion detection because false negatives represent attacks that remain undetected. Although class weighting improved Macro F1 in both linear classifiers, the associated reduction in Attack Recall may be undesirable in security-sensitive applications where attack detection is a primary objective. Accordingly, imbalance-handling strategies should be evaluated using class-specific metrics alongside aggregate measures such as accuracy and Macro F1. The suitability of a weighting strategy depends on the operational consequences of false positives and false negatives. Unlike oversampling approaches that introduce additional observations or undersampling approaches that remove existing observations, class weighting changes the relative contribution of classes during model training without requiring synthetic data generation or the removal of training records. This characteristic is compatible with the leakage-aware evaluation design used in this study because it allows imbalance handling to be examined without introducing synthetic observations into the training data. Nevertheless, the present findings establish the observed trade-off only for the two linear classifiers and configurations evaluated.

4.3 Comparative Performance and Random Forest

Among the five evaluated configurations, Random Forest achieved the highest reported accuracy, F1-score, and Macro F1 under both testing scenarios. On the leakage-controlled test set, it obtained an accuracy of 86.23%, an F1-score of 88.11%, a Macro F1-score of 85.88%, a ROC-AUC of 97.68%, and an average precision of 97.87%. Its Macro F1-score decreased from 0.8645 on the official test set to 0.8588 on the leakage-controlled test set, representing the smallest reported reduction among the evaluated configurations. The performance advantage of Random Forest suggests that its ensemble structure may be effective for modeling heterogeneous relationships among network traffic characteristics compared with the linear decision boundaries of Logistic Regression and Linear SVM. Previous intrusion detection research has also reported competitive performance from ensemble-based approaches (Das *et al.*, 2022). However, the present experiment compared a limited set of classifier configurations and did not directly investigate the mechanisms responsible for the observed performance differences. The findings therefore support the comparative advantage of Random Forest within the experimental setting rather than a general claim of superiority over other intrusion detection methods. The ablation results also show that Random Forest performance remained sensitive to the composition of the test set. Removing label-consistent overlapping records was associated with lower accuracy, F1-score, and Macro F1, whereas removing label-conflicting overlap was associated with modest improvements. Consequently, the relative stability of Random Forest under leakage-controlled evaluation should not be interpreted as immunity to cross-split feature overlap. Instead, it indicates that Random Forest experienced smaller reported performance reductions than the other evaluated configurations under the main comparison.

4.4 Security Implications and Remaining Challenges

From a security perspective, Random Forest achieved a reported Attack detection rate of 98.37% and a false negative rate of 1.63% on the leakage-controlled test set. The confusion matrix recorded 623 Attack observations misclassified as Normal out of 38,263 Attack records. This result indicates high sensitivity to Attack traffic, which is important because false negatives represent malicious activity that may remain undetected. However, the model also produced 9,540 false positives among 35,528 Normal records,

corresponding to a reported false positive rate of 26.85%. Thus, high attack detection sensitivity did not eliminate the potential operational burden of false alarms. In practical NIDS deployment, excessive false positives may increase analyst workload and complicate security monitoring. Further work could investigate threshold optimization, feature selection, cost-sensitive learning, and alternative ensemble configurations to examine whether specificity can be improved while maintaining an acceptable attack detection rate. These approaches are potential directions for future evaluation, rather than improvements demonstrated by the current experiment. The findings indicate that reliable intrusion detection evaluation should account for class imbalance and cross-split feature overlap. Evaluating classifiers exclusively on an official partition without auditing duplicated predictor patterns may produce a different performance estimate from evaluation on a test set containing only patterns absent from training. The ablation analysis also indicates that the observed effects depend on overlap type: label-consistent overlap was associated with more optimistic performance estimates, whereas removing label-conflicting overlap was associated with improved metrics. The leakage-aware evaluation framework therefore complements conventional classifier comparisons by examining performance sensitivity to exact feature-pattern overlap. These findings should be interpreted within the study's limitations. The evaluation used binary classification on a single benchmark dataset and focused on exact feature-pattern overlap. The results do not establish how the proposed procedure would perform across other datasets, temporal partitions, multiclass attack scenarios, or near-duplicate patterns. Further evaluation is needed before the findings can be generalized to operational network environments.

5. Conclusion

This study developed a leakage-aware evaluation framework for imbalanced network intrusion detection using the UNSW-NB15 dataset and compared Logistic Regression, Linear SVM, Random Forest, and class-weighted variants of the linear classifiers. The cross-split audit identified 1,302 unique overlapping predictor patterns affecting 8,541 test records (10.37% of the official test set). Of these patterns, 1,023 (78.57%) were label-consistent and 279 (21.43%) were label-conflicting, corresponding to 8,015 and 526 test records, respectively. Excluding all overlapping records yielded a leakage-controlled test set of 73,791 observations. The observed performance reductions across the evaluated classifiers indicate that results obtained from the official benchmark partition may differ from performance on test observations without exact predictor-pattern overlap with the training data. The ablation analysis revealed different evaluation outcomes when label-consistent and label-conflicting overlapping records were removed. Removing label-consistent records reduced Random Forest performance, whereas removing label-conflicting records produced modest improvements. These results suggest that the two types of overlap affect evaluation differently and that the overall performance change depends on the composition of the test set. Under leakage-controlled evaluation, Random Forest achieved an accuracy of 86.23%, an F1-score of 88.11%, a Macro F1-score of 85.88%, a ROC-AUC of 97.68%, and an average precision of 97.87%, outperforming the other evaluated configurations. Its attack recall was 98.37%, with a false negative rate of 1.63%. However, the false positive rate of 26.85% indicates that reducing false alarms remains an important operational challenge. Class weighting increased Macro F1 for both Logistic Regression and Linear SVM but reduced attack recall, highlighting a trade-off between class-balanced performance and attack detection sensitivity. These findings emphasize that intrusion detection models should be assessed using multiple complementary metrics, including class-specific errors, rather than aggregate performance alone. More broadly, reliable evaluation should consider class imbalance and train-test independence together to provide a more informative assessment of model performance on observations without exact predictor-pattern overlap. This study is limited to binary classification on a single benchmark dataset and an overlap audit based on exact predictor patterns. Consequently, its findings cannot be generalized directly to all operational network environments or other forms of data dependence. Future research should evaluate the proposed framework on additional intrusion detection datasets, temporally separated traffic, multiclass classification, and previously unseen attack scenarios. Further work should also investigate near-duplicate and similarity-based overlap detection, threshold calibration, feature selection, cost-sensitive learning, and alternative ensemble methods to reduce false positives while maintaining high attack detection sensitivity.

Acknowledgments

The authors thank the developers of the publicly available UNSW-NB15 dataset for making it accessible to the cybersecurity research community. The availability of this dataset has supported the comparative evaluation of machine learning classifiers and the investigation of data leakage in network intrusion detection. The authors also acknowledge the institutional support provided during the conduct of this study. Finally, the authors appreciate the contributions of researchers whose work has advanced the understanding of machine learning-based intrusion detection and informed the development of this study.

References

- Ahmad, R., Alsmadi, I., Al-Hamdani, W. A., & Tawalbeh, L. (2022). A deep learning ensemble approach to detecting unknown network attacks. *Journal of Information Security and Applications*, 67, 103196. <https://doi.org/10.1016/j.jisa.2022.103196>
- Ahmed, A., Asim, M., Ullah, I., Zainulabidin, & Ateya, A. A. (2024). An optimized ensemble model with advanced feature selection for network intrusion detection. *PeerJ Computer Science*, 10, e2472. <https://doi.org/10.7717/peerj-cs.2472>
- Alsaffar, A. M., Nouri-Baygi, M., & Zolbanin, H. M. (2024). Shielding networks: Enhancing intrusion detection with hybrid feature selection and stack ensemble learning. *Journal of Big Data*, 11(1), 133. <https://doi.org/10.1186/s40537-024-00994-7>
- Amien, J. Al, Sunanto, S., Rangkuti, M. A.-I., & Soni, S. (2025). Deteksi serangan dalam ekosistem IoT melalui analisis multi-class dengan model XGBoost dan penerapan teknik imbalance ratio pada dataset IoTID20. *Jurnal CoSciTech (Computer Science and Information Technology)*, 6(3), 568–574. <https://doi.org/10.37859/coscitech.v6i3.9861>
- Bouke, M. A., & Abdullah, A. (2023). An empirical study of pattern leakage impact during data preprocessing on machine learning-based intrusion detection models reliability. *Expert Systems with Applications*, 230, 120715. <https://doi.org/10.1016/j.eswa.2023.120715>
- Das, S., Saha, S., Priyoti, A. T., Roy, E. K., Sheldon, F. T., & Haque, A. (2022). Network intrusion detection and comparative analysis using ensemble machine learning and feature selection. *IEEE Transactions on Network and Service Management*, 19, 4821–4833. <https://doi.org/10.1109/TNSM.2021.3138457>
- Hore, S., Ghadermazi, J., Shah, A., & Bastian, N. D. (2024). A sequential deep learning framework for a robust and resilient network intrusion detection system. *Computers & Security*, 144, 103928. <https://doi.org/10.1016/j.cose.2024.103928>
- Joeres, R., Blumenthal, D. B., & Kalinina, O. V. (2025). Data splitting to avoid information leakage with DataSAIL. *Nature Communications*, 16(1), 3337. <https://doi.org/10.1038/s41467-025-58606-8>
- Layeghy, S., Baktashmotlagh, M., & Portmann, M. (2023). DI-NIDS: Domain invariant network intrusion detection system. *Knowledge-Based Systems*, 273, 110626. <https://doi.org/10.1016/j.knosys.2023.110626>
- Le, T.-T.-H., Shin, Y., Kim, M., & Kim, H. (2024). Towards unbalanced multiclass intrusion detection with hybrid sampling methods and ensemble classification. *Applied Soft Computing*, 157, 111517. <https://doi.org/10.1016/j.asoc.2024.111517>

- Li, X., Kong, K., Shen, H., Wei, Z., & Liao, X. (2024). Intrusion detection method based on imbalanced learning classification. *Journal of Experimental & Theoretical Artificial Intelligence*, 36(5), 657–677. <https://doi.org/10.1080/0952813X.2022.2104384>
- More, S., Idrissi, M., Mahmoud, H., & Asyhari, A. T. (2024). Enhanced intrusion detection systems performance with UNSW-NB15 data analysis. *Algorithms*, 17(2), 64. <https://doi.org/10.3390/a17020064>
- Rahma, F., Rachmadi, R. F., Pratomo, B. A., & Purnomo, M. H. (2023). Assessing the effectiveness of oversampling and undersampling techniques for intrusion detection on an imbalanced dataset. In *2023 IEEE Industrial Electronics and Applications Conference (IEACon)* (pp. 92–97). IEEE. <https://doi.org/10.1109/IEACon57683.2023.10370430>
- 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*. <https://doi.org/10.1109/TNSM.2024.3414305>
- Wijethilaka, R. W. K. S., Yapa, K., & Siriwardena, D. (2025). A cost effective machine learning based network intrusion detection system using Raspberry Pi for real time analysis. *PLOS ONE*, 20. <https://doi.org/10.1371/journal.pone.0331123>
- Xu, C., Zhan, Y. Z., Chen, G., Wang, Z., Liu, S., & Hu, W. (2025). Elevated few-shot network intrusion detection via self-attention mechanisms and iterative refinement. *PLOS ONE*, 20(1), e0317713. <https://doi.org/10.1371/journal.pone.0317713>