Matrik: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer
Vol. 25, No. 3, July 2026, pp. 491∼508
ISSN: 2476-9843, accredited by Kemenristekdikti, Decree No: 10/C/C3/DT.05.00/2025
DOI: 10.30812/matrik.v25i3.6147

Operational Weakness Mapping of Machine Learning–Based Intrusion Detection Systems under Realistic Deployment Scenarios

Fathoni Mahardika, Ema Utami, Kusrini, Ferry Wahyu Wibowo
Universitas AMIKOM Yogyakarta, Yogyakarta, Indonesia

Article Info

Article history:

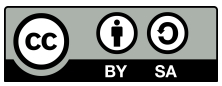
Received January 26, 2026
Revised May 28, 2026
Accepted June 03, 2026

Keywords:

Benchmarking;
Explainable artificial intelligence;
Intrusion detection system;
Machine learning;
Operational robustness.

ABSTRACT

As machine learning-based intrusion detection systems increasingly support information security risk management, prior systematic literature review findings indicate that many studies still emphasize benchmark accuracy while paying limited attention to robustness, interpretability, and operational feasibility. This study aims to map the operational weaknesses of machine learning-based intrusion detection systems under realistic deployment stressors. A directed replication and scenario-based stress-testing approach was applied using four public intrusion detection datasets, namely CICIDS2017, CICIDS2018, UNSW-NB15, and RanSMAP. The data were obtained from public repositories, converted to binary labels, cleaned by removing identifiers and non-numeric attributes, imputed with median values, scaled with MinMax normalization, and split into training and testing subsets. Supervised models, including Random Forest and XGBoost, were compared with unsupervised baselines, including Isolation Forest, LOF/kNN-distance, and DBSCAN, across scenarios covering baseline benchmarking, class imbalance, telemetry degradation, drift, parameter sensitivity, and micro-batch inference. The results show that supervised models achieved near-perfect baseline performance but degraded sharply under minor Gaussian noise, with F1-score dropping to 0.16 for Random Forest and 0.41 for XGBoost. Unsupervised models showed limited detection capability and high sensitivity to parameters. Although micro-batch inference achieved high throughput, alert burden remained a practical concern. These findings demonstrate that benchmark accuracy alone is insufficient for deployment readiness and that IDS evaluation should include robustness, interpretability, and alert-management analysis.



Corresponding Author:

Fathoni Mahardika,
Faculty of Computer Science, Informatics study program,
Universitas AMIKOM Yogyakarta, Yogyakarta, Indonesia,
Email: fathonimahardika@students.amikom.ac.id

How to Cite:
Fathoni Mahardika, Ema Utami, Kusrini, and Ferry Wahyu Wibowo, ”Operational Weakness Mapping of Machine Learning–Based Intrusion Detection Systems under Realistic Deployment Scenarios”, *MATRIK: Jurnal Manajemen, Teknik Informatika, dan Rekayasa Komputer*, Vol. 25, No. 3, pp. 491-508, July, 2026.

1. INTRODUCTION

The rapid expansion of network-connected systems and Internet of Things (IoT) devices has significantly increased the global digital footprint and expanded the potential cyberattack surface. As organizations integrate smart devices and distributed infrastructures into their operations, their exposure to sophisticated cyber threats grows proportionally. Recent studies highlight that the heterogeneity and scale of modern network ecosystems make conventional security mechanisms increasingly insufficient, thereby requiring adaptive and intelligent protection systems capable of analyzing complex network behaviors in near real time [1, 2]. In this context, Intrusion Detection Systems (IDS) play a critical role by continuously monitoring network traffic and identifying potentially malicious activities.

The urgency of this research arises from the growing dependence of organizations, including higher education institutions, on interconnected digital services, cloud platforms, and heterogeneous network infrastructures. These environments generate large volumes of security telemetry, but the quality of such telemetry is often affected by missing values, noise, incomplete packet information, and changing traffic patterns. In this condition, intrusion detection systems that perform well on clean benchmark datasets may fail when deployed in real operational environments. Therefore, evaluating machine learning-based IDS only using conventional benchmark accuracy is no longer sufficient for information security risk management.

This concern is consistent with the authors' previous systematic literature review (SLR), which found that machine learning-based information security risk management (ISRM) studies remain fragmented, underexplored in higher education contexts, and limited in terms of interpretability, privacy-aware design, and operational validation [3, 4]. Therefore, this study is developed as an empirical follow-up to evaluate the operational robustness and deployment feasibility of machine learning-based intrusion detection systems under realistic conditions.

Traditional IDS technologies primarily rely on signature-based detection, which compares network traffic against known attack patterns. Although effective for previously observed threats, such systems cannot detect novel or zero-day attacks without predefined signatures [5]. To address this limitation, anomaly-based IDS approaches have been developed to model normal network behavior and identify deviations that may indicate intrusions. Machine learning (ML) techniques have been widely applied in this area because they can automatically learn discriminative patterns from data and adapt to evolving threat behaviors without relying on manually defined rules [6, 7]. Prior studies also report that ML algorithms such as Random Forest, XGBoost, Isolation Forest, DBSCAN, and K-Nearest Neighbor can achieve strong benchmark performance in distinguishing benign and malicious activities under curated datasets [2, 6, 8–10].

Previous research has demonstrated that ML algorithms such as Random Forest, XGBoost, Isolation Forest, DBSCAN, and other classical models can achieve strong benchmark performance when evaluated on curated datasets including CICIDS2017, CICIDS2018, and UNSW-NB15 [2, 6, 8–10]. However, despite these promising results, deploying ML-based IDS in operational environments remains challenging. Issues such as noisy telemetry, incomplete data, temporal drift, and the lack of interpretability can significantly affect system reliability. In addition, many high-performing models behave as "black boxes," limiting analysts' ability to understand and justify detection outcomes. Explainable Artificial Intelligence (XAI) methods, particularly SHAP and LIME, have therefore been proposed to improve transparency by attributing model predictions to feature-level contributions [11–13].

Existing literature reveals several important gaps in operational readiness. Early survey studies provide broad overviews of machine learning and deep learning approaches for intrusion detection but offer limited evaluation under realistic operational stress conditions and do not incorporate structured scenario-based testing frameworks [6]. Subsequent review studies similarly emphasize performance improvements while giving less attention to robustness, alert burden, and practical deployment constraints [14]. More recent works that integrate explainable artificial intelligence focus primarily on improving interpretability, yet often do not jointly evaluate robustness and alert feasibility. Conversely, studies that investigate robustness tend to assess model stability under specific perturbations but rarely translate these findings into systematic mappings of operational weaknesses or actionable decision-support insights.

Another important concern is robustness. Most IDS evaluations rely on clean and balanced benchmark datasets that may not accurately represent real-world network conditions. As a result, models that demonstrate near-perfect accuracy in experimental settings may exhibit fragile behavior when exposed to noisy data, missing telemetry, or changing traffic patterns. This discrepancy highlights the need for evaluation frameworks that simulate realistic deployment conditions and reveal operational weaknesses beyond conventional accuracy metrics [1, 9]. To address this gap, recent studies increasingly explore hybrid anomaly detection frameworks that combine unsupervised and supervised paradigms with feature engineering to improve robustness and generalization performance across datasets [15]. Other works study resilience against adversarial inputs, including GAN-generated perturbations and security-focused DL architectures that improve robustness in complex scenarios [16, 17]. Related methodological insights can also be drawn from hybrid AI–ML models in other risk domains, where feature fusion, interpretability, and overfitting–underfitting trade-offs are central concerns [18]. Additionally, semi-supervised learning and clustering-based approaches have been reported to

improve detection performance in low-label and heterogeneous environments [19, 20].

Therefore, the objective of this study is to empirically map the operational weaknesses of machine learning-based IDS under realistic deployment stressors derived from the research gaps identified in previous SLR findings. The main contributions of this study are threefold. First, this study translates SLR-based research gaps into an empirical, scenario-based evaluation framework that covers robustness, interpretability, and near-real-time feasibility. Second, it compares supervised and unsupervised IDS models not as a leaderboard, but as an operational weakness mapping across false-negative risk, false-positive burden, drift, parameter sensitivity, and alert rate. Third, it provides deployment-oriented insights showing that near-perfect benchmark performance may still be fragile under telemetry degradation.

2. RESEARCH METHOD

2.1. Research Design

This study employs a directed replication and scenario-based stress-testing approach to evaluate the operational weaknesses of machine learning-based intrusion detection systems (IDS). The research is developed as an empirical follow-up to the authors' previous systematic literature review (SLR), which identified limited operational validation in prior ML-based information security risk management studies [1]. Unlike conventional benchmark-oriented evaluations, this study focuses on identifying failure modes that may affect operational usability, including missed attacks (false negatives), excessive false alarms (false positives), instability under perturbations, and near-real-time feasibility constraints.

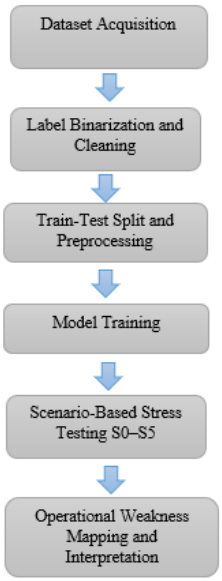


Figure 1. Research Method

As illustrated in Figure 1, the research workflow begins with preprocessing the dataset and separating training and testing data to ensure the development of a reliable model. The next stage involves training the base model, followed by scenario-based evaluations (S0–S5) under various operational conditions. These evaluations include robustness testing, parameter sensitivity analysis, and near-real-time inference assessments to measure the model's stability and efficiency. Additionally, interpretability analysis is applied to identify the variables most influential on prediction outcomes. Finally, the workflow concludes with iterative operational risk mapping to generate application-oriented insights and practical implementation recommendations.

2.2. Dataset Acquisition

Four publicly available intrusion detection datasets were obtained from public repositories: CICIDS2017, CICIDS2018, UNSW-NB15, and RanSMAP [1, 21]. These datasets were selected because they represent a wide range of network traffic char-

acteristics, attack scenarios, and operational environments commonly used in intrusion detection research. Additionally, each dataset contains distinct distributions of normal and malicious traffic, thereby facilitating a comprehensive evaluation of the model under various conditions. The use of multiple datasets also enhances the robustness and generalization capabilities of the proposed framework. Furthermore, the combination of these reference datasets supports comparative analysis and reduces the risk of bias associated with specific datasets in experimental results.

2.3. Data Input and Label Construction

The datasets were imported into the experimental pipeline and transformed into a binary classification format. BENIGN traffic was labeled as 0, while Attack/Anomaly traffic was labeled as 1. If timestamp attributes were available, they were retained to support time-based evaluation in the drift scenario (S3). Otherwise, row order was used as a temporal proxy. To support consistent evaluation, stratified train–test splitting was applied using an 80:20 ratio with a fixed random_state to preserve class distribution consistency between training and testing data.

2.4. Data Preprocessing

A standardized numeric preprocessing pipeline was applied consistently across all datasets. The preprocessing steps included column name normalization, replacing ±inf values with NaN, removing identifier and non-numeric/string attributes such as IP address fields, median imputation for missing values, and MinMax normalization to scale all features to the range [0,1]. To prevent data leakage, all preprocessing objects, including the SimpleImputer and MinMaxScaler, were fitted only on the training data and reused for testing and scenario evaluations. The same preprocessing configuration was maintained across all operational scenarios to isolate scenario-specific effects. Feature correlation analysis was additionally conducted using a correlation threshold ≥ 0.98 to identify highly redundant features and detect potential implicit target leakage.

To evaluate cross-dataset generalization, supervised models trained on CICIDS2018 were tested on the UNSW-NB15 dataset via semantic feature alignment. Corresponding features such as dur $\rightarrow$ flow_duration, spkts $\rightarrow$ tot_fwd_pkts, and sbytes $\rightarrow$ totlen_fwd_pkts were mapped to maintain dimensional consistency. In addition, unmatched features were filled with default values (0.0) to avoid inconsistencies during model inference. The feature ordering was then adjusted to match the original training structure exactly, ensuring compatibility with the trained models. This alignment process enabled the evaluation of model transferability and robustness across heterogeneous network traffic environments.

2.5. Model Development

This study evaluates five baseline algorithms representing different intrusion detection paradigms [9, 15, 22, 23]. The supervised learning category includes Random Forest (RF) and XGBoost, as both algorithms have demonstrated strong performance in intrusion classification tasks [8–10, 22, 23]. Random Forest combines predictions from multiple decision trees using majority voting, while XGBoost constructs additive tree ensembles with regularization to improve classification performance and reduce overfitting. The RF prediction mechanism is formally expressed in Equation 1, where the final class label $\hat{y}$ is determined as the mode of predictions from B individual decision trees $T_1(x)$ through TB(x).

Lightweight hyperparameter configurations were applied to maintain computational efficiency during repeated scenario evaluations. In addition, limited hyperparameter optimization was conducted using RandomizedSearchCV (n_iter = 5, cv = 3, scoring = F1) to provide a more calibrated comparison without extensive tuning. This approach reflects realistic deployment conditions where computational resources and tuning time are often limited [8, 9]. Furthermore, the lightweight configuration strategy reduces training overhead while maintaining acceptable predictive performance across diverse experimental scenarios. As a result, the proposed framework remains suitable for practical intrusion detection environments requiring efficient and scalable model deployment.

To represent label-scarce operational environments, this study also employs unsupervised anomaly detection approaches [15]. Isolation Forest (IF) detects anomalies by partitioning data randomly and isolating anomalies [24]. Local Outlier Factor (LOF), combined with kNN distance, is a distance-based anomaly detection method that evaluates local density differences among neighboring samples [25]. In addition, DBSCAN is used as a density-based clustering method with adaptive ε estimation and MinPts sensitivity analysis [23]. These algorithms were selected not to establish benchmark superiority, but to expose distinct operational weak points related to robustness, parameter sensitivity, density instability, and alert volatility under realistic deployment conditions.

2.6. Scenario-Based Evaluation

Operational evaluations were conducted through six scenarios (S0–S5) designed to simulate realistic deployment conditions. Each scenario represents a different experimental setting, including baseline performance evaluation, robustness assessment, parameter sensitivity analysis, and near-real-time inference testing. These scenario-based evaluations were designed to examine the adaptability and operational feasibility of the proposed framework under varying network environments. Collectively, the six scenarios cover the full range of operational stressors identified in prior SLR findings, including class imbalance, telemetry degradation, temporal drift, parameter brittleness, and latency constraints. The detailed description of each evaluation scenario is presented in the following table Table 1.

Table 1. Scenario-Based Operational Evaluation Design

Scenario	Purpose	Procedure
S0	Baseline benchmark	Stratified 80:20 train–test split
S1	Class imbalance	Train undersampling from 1:1 to 1:100
S2	Telemetry degradation	Test-only Gaussian noise and missingness
S3	Drift/shift	Time-based split and rolling-window evaluation
S4	Parameter sensitivity	Sweep ε, MinPts, contamination, and k
S5	Near real-time feasibility	Micro-batch latency and throughput evaluation

Operational evaluations were performed using six scenarios (S0–S5) to represent realistic deployment conditions, as summarized in Table 1. These scenarios included baseline benchmarking, class imbalance simulation, telemetry degradation via Gaussian noise and missing values, drift or shift evaluation using time-based and rolling-window approaches, parameter sensitivity analysis, and near-real-time feasibility assessment via latency and throughput measurements. In the telemetry degradation scenario (S2), Gaussian noise was injected into normalized numeric features without retraining the model.

In the telemetry degradation scenario (S2), Gaussian noise was injected into normalized numeric features without retraining the model. This scenario was designed to simulate sensor instability, transmission interference, and imperfect telemetry conditions commonly encountered in real-world network environments. By introducing controlled perturbations into the input data, the experiment evaluated the robustness and stability of the trained models under degraded operational conditions. The noise injection process is mathematically expressed in the following equation.

$$x' = clip(x + \varepsilon, 0, 1), \varepsilon \sim N(0, \sigma^2) \tag{1}$$

Where x denotes the normalized input feature vector, while x' denotes the feature vector after being subjected to disturbance or Gaussian noise injection. The variable ε represents Gaussian noise drawn from a normal distribution with variance σ^2. The clip (') function is used to constrain the feature values to remain within the normalization range [0,1]. The formulation in Equation 1 is used in the telemetry degradation scenario (S2) to simulate sensor interference and data instability during inference. This approach aims to evaluate the model's robustness against input quality degradation in realistic operational environments.

2.7. Interpretability Analysis

SHAP was used to quantify both global feature importance and local explanations for representative false-positive and false-negative cases. This interpretability analysis improves model transparency and supports the practical deployment of intrusion detection systems in operational environments. Global SHAP importance was calculated using mean absolute SHAP values across samples to identify the most influential features contributing to model predictions. The mathematical formulation of global SHAP importance is presented in the following equation.

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!\,(|N| - |S| - 1)!}{|N|!} \left[f(S \cup \{i\}) - f(S) \right] \tag{2}$$

In Equation 2, ϕ_i represents the contribution of the i-th feature to the model's decision based on the SHAP (SHapley Additive exPlanations) method. The symbol S denotes the subset of features excluding feature i, while N is the set of all features used in the model. The function f(S) represents the model's output based on a specific feature subset, while f(S∪ {i}) indicates the prediction after feature i is added to that subset. The global SHAP value is then calculated as the absolute average of SHAP values across all samples, yielding a stable and easily interpretable feature ranking.

2.8. Evaluation Metrics

Performance evaluation used Accuracy, Precision, Recall, F1-score, and PR-AUC as the primary metrics under imbalanced conditions [8, 10]. Operational usability was additionally evaluated using False Alarm Rate (FAR) and alert_rate to assess the practicality of the proposed framework in real deployment environments. These metrics were selected to provide a comprehensive assessment of both predictive performance and operational reliability. Furthermore, the combination of classification and operational metrics enables a more comprehensive analysis of detection capability, false-alarm behavior, and system responsiveness under varying network conditions. The evaluation metrics are mathematically defined as follows.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3}$$

$$Precision = \frac{TP}{TP + FP} \tag{4}$$

$$Recall = \frac{TP}{TP + FN} \tag{5}$$

$$F_1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{6}$$

$$FAR = \frac{FP}{FP + TN} \tag{7}$$

In Equations 3, 4, 5, 6, and 7, TP (True Positive) indicates the amount of attack traffic that was correctly classified as an attack. TN (True Negative) indicates the number of normal traffic successfully identified as benign. FP (False Positive) represents normal traffic incorrectly classified as an attack, while FN (False Negative) indicates attack traffic that the system failed to detect. Accuracy is used to measure the overall classification accuracy, while Precision, Recall, and F1-score are used to evaluate the model's detection capability under imbalanced data conditions. Additionally, FAR (False Alarm Rate) is used to measure the proportion of false alarms generated by the system across all benign traffic.

3. RESULT AND ANALYSIS

3.1. Data Validation and Pipeline (Leakage, Correlation, Generalization)

First, a data leakage check was performed to ensure that the entire modeling process was not contaminated by information from the test data. The preprocessing pipeline in this study was explicitly designed to prevent leakage through several key mechanisms. Preprocessing objects such as SimpleImputer (median strategy) and MinMaxScaler are only fit to the training data (fit-on-train-only), while transformations on the test data are performed using parameters learned from the training data. Additionally, in the perturbation scenario (S2), perturbations in the form of noise and missing data are injected only into the test data after the model has finished training (test-only perturbation), so they do not affect the model's learning process. Data splitting is also performed using an 80:20 stratified split with a fixed random_state to maintain consistency in class distribution between the training and test data.

As an additional verification step, automatic feature leakage detection is performed through correlation analysis between features and the target variable using a strict threshold ($|r| > 0.98$). This process involves calculating the Pearson correlation matrix for all preprocessed features, identifying features with high absolute correlation to the target as leakage candidates, and revalidating to ensure no administrative or identity-based features (such as flow_id, source_ip, and destination_ip) are present in the feature set.

The inspection results show that no features with a correlation greater than 0.98 with the target were found, with the system output being "No extreme feature-to-target leakage detected (> 0.98 correlation)". Additionally, attributes that could potentially cause leakage, such as flow_id, source_ip, destination_ip, simillarhttp, and timestamp, were eliminated during the initial cleaning stage. These findings indicate that there is no evidence of data leakage in the pipeline, so the model evaluation results can be considered valid and not influenced by bias due to unintended information.

Second, a feature correlation analysis was conducted to understand the relationships between features and their contributions to the target variable. This analysis uses Pearson's correlation to identify the most informative features for intrusion detection. The results are visualized via a correlation heatmap displaying the 20 features with the highest correlation to the target, along with feature

rankings based on their absolute correlation values against the is_attack variable. This analysis provides an initial understanding of feature relevance and helps support the feature selection process before model training.

Key findings indicate that the correlation patterns align with SHAP-based interpretability results, where features related to port and segment characteristics (such as fwd_seg_size_min and dst_port), throughput indicators (such as 'bwd_pkts/s', 'flow_pkts/s', and 'fwd_pkts/s'), as well as TCP-based attributes like 'init_fwd_win_byts', 'init_bwd_win_byts', 'psh_flag_cnt', and 'ack_flag_cnt', are dominant factors in the model's decision-making. This reliance on packet- and time-based aggregation features also explains the model's sensitivity to telemetry degradation scenarios (S2), as these features are highly susceptible to disturbances such as noise and missing data.

Third, model generalization was tested via cross-dataset validation to evaluate the model's ability to adapt to different data domains. In this test, supervised models (XGBoost and Random Forest) trained on the CICIDS2018 dataset were evaluated on the UNSW-NB15 dataset. Given the differences in feature structures between datasets, a feature alignment process was performed through semantic mapping between features (e.g., dur to flow_duration, spkts to tot_fwd_pkts, and sbytes to totlen_fwd_pkts). Features without a corresponding counterpart were filled with a value of 0.0 to maintain input dimension consistency, and the column order was adjusted to match the training schema exactly.

The evaluation results show that both models experienced a very significant drop in performance on the target dataset, with recall and F1-score values of 0, although accuracy remained in the range of 0.55. As shown in Table 2, the models failed to correctly identify any positive (attack) instances under cross-dataset conditions, despite maintaining moderate accuracy due to class distribution effects. These findings suggest that the learned feature representations from the source dataset were insufficiently transferable to the target dataset due to differences in traffic characteristics and attack patterns.

Table 2. Evaluation of Model Generalization in Cross-Dataset Scenarios (Domain Shift)

Model	Accuracy	Precision	Recall	F1-Score
XGBoost	∼0.55	0.0	0.0	0.0
RandomForest	∼0.55	0.0	0.0	0.0

These results indicate that the model failed to detect anomalies in the new domain, suggesting significant differences in data distribution between the datasets. Further analysis of the cross-dataset validation results revealed the presence of an extreme domain shift phenomenon, which was the primary cause of the drastic drop in performance (recall = 0% for both models). Visualization of the distribution using Kernel Density Estimation (KDE) on the aligned features shows that the distributions between CICIDS2018 and UNSW-NB15 are almost entirely disjoint. Key features such as flow_duration and tot_fwd_pkts have vastly different value ranges across the two datasets, making the patterns learned in the source domain irrelevant to the target domain.

Consequently, the decision boundary learned by the model on CICIDS2018 becomes invalid when applied to UNSW-NB15. The model associates specific value ranges with attack activity; however, in the target dataset, those ranges either do not appear or are associated with normal traffic, causing the model to fail to detect anomalies. This finding is reinforced by an analysis of the drift score (normalized difference in means), which shows very high values (over 67 for some features), indicating an extreme shift in distribution between domains.

To address this issue, several mitigation efforts were undertaken in stages. First, hyperparameter optimization was performed using RandomizedSearchCV to adjust the model configuration to the target domain. However, evaluation results still showed a recall of 0%, indicating that the problem did not lie in the model parameters. Second, renormalization was performed using StandardScaler to align the scales of the distributions between the training and testing data. This approach also did not yield any improvement, as the differences in distribution were structural in nature, not merely a matter of scale. Third, domain-invariant feature selection was performed by selecting features with the lowest drift score (i.e., the most stable across datasets). Nevertheless, the results remained unchanged, with recall still at 0%.

3.2. Baseline Performance (S0)

We executed this research benchmark pipeline consistently across four public IDS datasets (CICIDS2018, CICIDS2017, UNSW-NB15, and RanSMAP) to assess model behavior under comparable preprocessing and evaluation settings. The evaluation metrics defined in Equations 3, 4, 5, 6, and 7 were used to calculate the baseline performance reported in Table 3. As an anchor for the directed replication, Table 3 reports the S0 baseline metrics on the CICIDS2018 subset used in the main experiments, while scenario-level outputs are saved as CSV artifacts for cross-dataset comparison. This standardized evaluation procedure enables fair assessment of model robustness, consistency, and generalization across heterogeneous intrusion detection datasets.

Table 3. Baseline performance (CICIDS2018 subset)

Model	PRAUC	Precision	Recall	F1	FAR	AlertRate
XGBoost	1.000.000	0.999972	1.000.000	0.999986	0.000016	0.362880
RandomForest	1.000.000	1.000.000	1.000.000	1.000.000	0.000000	0.362870
IsolationForest	0.257198	0.000000	0.000000	0.000000	0.129220	0.082330
LOF	0.257732	0.040300	0.011106	0.017413	0.150629	0.100000
DBSCAN	-	0.000000	0.000000	0.000000	0.001209	0.000770

Table 3 shows baseline performance. Supervised models reach near-perfect PR-AUC/F1 with extremely low FAR, while in the evaluated feature space and calibration setting, unsupervised models did not provide sufficient separation between benign and attack traffic, suggesting limited suitability as standalone detectors under the tested operational conditions. The near-perfect baseline performance of Random Forest and XGBoost is consistent with prior IDS studies reporting strong performance of ensemble and hybrid models on structured cybersecurity datasets [26]. For example, the authors' previous SLR found that Random Forest, Gradient Boosting, Autoencoder-XGBoost, and Deep Autoencoder-Random Forest models frequently achieved high accuracy in IDS and ISRM-related tasks [4]. However, unlike these prior studies, the present result should not be interpreted as final deployment readiness because subsequent stress scenarios reveal substantial performance degradation under telemetry noise.

3.3. Imbalance Robustness (S1)

In S1 imbalance robustness, supervised retraining shows metric stability even though the anomaly ratios are made rarer in the training data. This finding is important because operational intrusion detection systems are often characterized by highly imbalanced traffic distributions, in which malicious activity occurs far less frequently than normal traffic. The stable performance across imbalanced conditions indicates that the proposed models are capable of maintaining reliable detection capability despite limited attack representation.

Table 4. S1 Imbalance (Train Undersampling; Test Unchanged)

TrainRatio	Model	PRAUC	Precision	Recall	F1	FAR	AlertRate
1.00	RandomForest	1.000.000	0.999986	1.000.000	0.999993	0.000008	0.362876
1.00	XGBoost	1.000.000	1.000.000	1.000.000	1.000.000	0.000000	0.362872
0.10	RandomForest	1.000.000	1.000.000	1.000.000	1.000.000	0.000000	0.362870
0.10	XGBoost	1.000.000	0.999972	1.000.000	0.999986	0.000016	0.362880
0.02	RandomForest	1.000.000	1.000.000	1.000.000	1.000.000	0.000000	0.362870
0.02	XGBoost	1.000.000	0.999972	1.000.000	0.999986	0.000016	0.362880
0.01	RandomForest	1.000.000	1.000.000	0.999604	0.999802	0.000000	0.362727
0.01	XGBoost	1.000.000	0.999972	0.999604	0.999788	0.000016	0.362743

As shown in Table 4, both Random Forest and XGBoost maintain near-perfect PRAUC, precision, recall, and F1 scores across all train ratios (from 1.00 down to 0.01), with only a very slight decrease in recall and F1 at the most extreme imbalance level. Additionally, FAR remains close to zero, and the alert rate stays stable around 0.36, indicating that model performance is largely unaffected by severe class imbalance during training and demonstrating robustness in this scenario. The stability under class imbalance supports prior findings that ensemble models can maintain strong classification performance on structured IDS datasets [27]. However, this study extends prior work by showing that robustness to class imbalance does not necessarily imply robustness to degraded telemetry, as demonstrated later in S2.

3.4. Noise and Missing-Data Robustness (S2)

S2 is the most "operational" weakness found: telemetry degradation (noise) causes recall/F1 to fall sharply even when S0 is perfect. Missingness is lighter because the imputer can still recover some information. The telemetry degradation process in this scenario follows Equation 1. The patterns in Table 5 clearly illustrate this contrast: under noise perturbations, both Random Forest and XGBoost experience drastic drops in recall and F1-score, with Random Forest recall falling to as low as 0.045994 and XGBoost to around 0.25, even as precision remains close to 1. This imbalance indicates that the models become overly conservative, detecting very few attacks while appearing superficially accurate. In comparison, missing-data scenarios produce only modest degradation, as the imputation process preserves much of the underlying signal, resulting in relatively stable performance. In real-world operations,

small sensor/telemetry distortions can convert a ”perfect” benchmark IDS into a high-FN risk (missed attacks), which directly harms risk decisions.

Table 5. S2 Robustness (Test-Only Perturbation; Trained Once On S0)

Model	Condition	PRAUC	Precision	Recall	F1	FAR	AlertRate
RandomForest	Baseline (S0)	1.000000	1.000000	1.000000	1.000000	0.000000	0.362870
RandomForest	Noise σ=0.01	0.412572	1.000000	0.090749	0.166397	0.000000	0.032931
RandomForest	Noise σ=0.05	0.389905	0.998803	0.045994	0.087939	0.000016	0.016717
RandomForest	Missing r=0.01	0.999954	1.000000	0.994351	0.997167	0.000000	0.360822
RandomForest	Missing r=0.05	0.999760	1.000000	0.964891	0.982132	0.000000	0.349600
XGBoost	Baseline (S0)	1.000000	0.999972	1.000000	0.999986	0.000016	0.362880
XGBoost	Noise σ=0.01	0.413140	0.998288	0.257144	0.408949	0.000016	0.093208
XGBoost	Noise σ=0.05	0.392484	0.997160	0.251578	0.401787	0.000032	0.091010
XGBoost	Missing r=0.01	0.999954	0.999972	0.979139	0.989446	0.000016	0.355325
XGBoost	Missing r=0.05	0.999763	0.999969	0.902775	0.948890	0.000016	0.327825

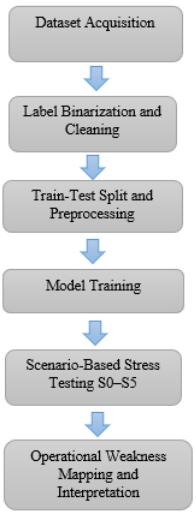


Figure 2. F1 degradation under inference-time Gaussian noise for Random Forest and XGBoost

Figure 2 visualizes the brittleness observed in Table 5 by plotting F1 as a function of Gaussian noise level (σ). Even small perturbations (σ=0.01) sharply reduce recall-driven performance, indicating that benchmark separability does not translate into operational reliability under telemetry degradation. Figure 2 illustrates the degradation of F1 scores under inference-time Gaussian noise. While both models achieve near-perfect performance in the baseline condition ($\sigma = 0$), even small perturbations ($\sigma = 0.01$) cause a sharp drop in recall-driven performance, particularly for Random Forest. This result highlights that strong benchmark separability does not necessarily translate into operational robustness under degraded telemetry conditions.

Unlike prior benchmark-oriented studies that mainly present static accuracy, precision, recall, or F1-score under clean test conditions, Figure 2 visualizes model behavior under inference-time telemetry degradation. This figure therefore shifts the interpretation from ”which model is most accurate” to ”which model remains reliable when input quality deteriorates.” The sharp decline in F1-score shows that robustness testing provides additional evidence not captured by conventional benchmark tables. Furthermore, the results highlight the sensitivity of intrusion detection models to noisy or degraded telemetry inputs, which are common in operational environments.

3.5. Shift/Drift and Near Real-time Scenarios (S3–5)

Here are the rolling-window results (CICIDS2018 example) demonstrating model consistency under changing traffic patterns and temporal distribution shifts. This evaluation was conducted to assess the stability of model performance across sequential data

segments simulating evolving operational conditions. The results provide insight into the adaptability and robustness of the proposed intrusion detection framework in the face of dynamic network behavior over time. The complete evaluation results are presented in the following table:

Table 6. S3 Shift/Drift (Rolling Windows on Test)

Window	Model	PRAUC	Precision	Recall	F1	FAR	AlertRate
1	RandomForest	1.000.000	1.000.000	1.000.000	1.000.000	0.000000	0.289133
1	XGBoost	1.000.000	1.000.000	1.000.000	1.000.000	0.000000	0.289133
2	RandomForest	1.000.000	0.999968	1.000.000	0.999984	0.000016	0.345127
2	XGBoost	1.000.000	0.999968	1.000.000	0.999984	0.000016	0.345127
3	RandomForest	1.000.000	0.999965	1.000.000	0.999982	0.000016	0.290897
3	XGBoost	1.000.000	0.999965	1.000.000	0.999982	0.000016	0.290897
4	RandomForest	1.000.000	0.999976	1.000.000	0.999988	0.000016	0.405093
4	XGBoost	1.000.000	0.999976	1.000.000	0.999988	0.000016	0.405093

The detailed results in Table 6 show that both Random Forest and XGBoost maintain highly stable performance across all windows, with PRAUC consistently at 1.0 and only negligible variations in precision, recall, and F1-score. This indicates strong resilience to moderate distribution shifts over time. However, the alert rate varies noticeably across windows (ranging from approximately 0.29 to 0.40), suggesting that while detection capability remains stable, the volume of alerts fluctuates with the underlying traffic pattern.

The S4 results highlight key operational weaknesses in density- and distance-based methods, where small changes in parameters can lead to significant variations in the False Alarm Rate (FAR) and alert rate. This sensitivity indicates that performance stability is highly dependent on parameter configuration, which could potentially complicate deployment in dynamic production environments. In practical intrusion detection systems, unstable parameter behavior may increase operational overhead due to frequent recalibration and monitoring requirements. Furthermore, these findings suggest that robust parameter selection is essential for maintaining reliable detection performance and minimizing excessive false alerts in real-world network operations.

Table 7. S4 Parameter Sensitivity

Setting	PRAUC	Precision	Recall	F1	FAR	AlertRate
DBSCAN_Eps0.75_Min10	-	0.000000	0.000000	0.000000	0.030663	0.019528
DBSCAN_Eps0.75_Min20	-	0.000000	0.000000	0.000000	0.015179	0.009665
DBSCAN_Eps1.0_Min10	-	0.000000	0.000000	0.000000	0.001209	0.000770
DBSCAN_Eps1.0_Min20	-	0.000000	0.000000	0.000000	0.000661	0.000421
DBSCAN_Eps1.25_Min10	-	0.000000	0.000000	0.000000	0.000221	0.000141
DBSCAN_Eps1.25_Min20	-	0.000000	0.000000	0.000000	0.000141	0.000090
IsolationForest_Cont0.01_Est100	0.257198	0.000000	0.000000	0.000000	0.129220	0.082330
IsolationForest_Cont0.05_Est100	0.257198	0.000000	0.000000	0.000000	0.129220	0.082330
LOF_k10_Th95	0.257732	0.040300	0.011106	0.017413	0.150629	0.100000
LOF_k10_Th99	0.257732	0.029061	0.005553	0.009321	0.149964	0.099557
LOF_k20_Th95	0.257732	0.040300	0.011106	0.017413	0.150629	0.100000
LOF_k20_Th99	0.257732	0.029061	0.005553	0.009321	0.149964	0.099557

Table 8. S4 Volatility Summary (Range)

Family	n	FAR(min)	FAR(max)	Alert(min)	Alert(max)	F1(min)	F1(max)
DBSCAN	6	0.000141	0.030663	0.000090	0.019528	0.000000	0.000000
IsolationForest	2	0.129220	0.129220	0.082330	0.082330	0.000000	0.000000
LOF	4	0.149964	0.150629	0.099557	0.100000	0.009321	0.017413

Based on Table 7 and Table 8, DBSCAN exhibits the highest volatility, with a very wide range of FAR values resulting from small parameter changes, while Isolation Forest demonstrates stable performance but with very low detection capability. On the other hand, LOF provides a slight improvement in F1 but is still accompanied by a high FAR; thus, overall, density- and distance-based methods demonstrate limitations in stability and detection effectiveness for operational use.

Next, the evaluation shifts to the S5 scenario, which focuses on near-real-time inference performance using a micro-batch approach to assess the balance between detection accuracy and system efficiency. This scenario was designed to simulate operational intrusion detection conditions where incoming network traffic must be processed continuously with minimal latency. The evaluation emphasizes the proposed framework's ability to maintain reliable predictive performance while handling streaming-like data-processing constraints.

Table 9. S5 Near Real-Time (Micro-Batch) Inference

BatchSize	Model	PRAUC	Precision	Recall	F1	FAR	AlertRate	Mean_Latency	P95_Latency	Throughput
5000	RandomForest	1.000.000	1.000.000	1.000.000	1.000.000	0.000000	0.362870	0.051522	0.054926	96952.19
5000	XGBoost	1.000.000	0.999972	1.000.000	0.999986	0.000016	0.362880	0.049024	0.060454	101991.84
20000	RandomForest	1.000.000	1.000.000	1.000.000	1.000.000	0.000000	0.362870	0.058361	0.061701	342724.23
20000	XGBoost	1.000.000	0.999972	1.000.000	0.999986	0.000016	0.362880	0.057925	0.065860	345366.09

Table 10. S5 Cross-Dataset Micro-Batch Summary

Dataset	Model	BatchSize	PRAUC	Precision	Recall	F1	FAR	AlertRate	MeanLatency	Throughput
RanSMAP	RandomForest	5000	0.999946	0.999432	0.999983	0.999707	0.000017	0.823083	0.041003	122135.70
RanSMAP	XGBoost	5000	0.999741	0.998992	0.999950	0.999471	0.000030	0.823238	0.038279	130589.95
RanSMAP	RandomForest	20000	0.999946	0.999432	0.999983	0.999707	0.000017	0.823083	0.046301	431944.98
RanSMAP	XGBoost	20000	0.999741	0.998992	0.999950	0.999471	0.000030	0.823238	0.047527	426888.67
UNSW-NB15	RandomForest	5000	0.983180	0.965829	0.944756	0.955176	0.018601	0.500143	0.099301	50345.15
UNSW-NB15	XGBoost	5000	0.985926	0.968817	0.947148	0.957860	0.017521	0.499101	0.093311	53514.65
UNSW-NB15	RandomForest	20000	0.983180	0.965829	0.944756	0.955176	0.018601	0.500143	0.112355	177973.09
UNSW-NB15	XGBoost	20000	0.985926	0.968817	0.947148	0.957860	0.017521	0.499101	0.105595	189410.63

Based on Table 9 and Table 10, both Random Forest and XGBoost maintain near-perfect detection performance in micro-batch settings, with PRAUC, precision, recall, and F1 remaining close to 1.0 while achieving low latency and high throughput, especially at larger batch sizes. However, although computational efficiency is achieved, the alert rate remains relatively high (around 0.36 in Table 9 and even higher in RanSMAP in Table 10), indicating that a significant portion of traffic is flagged. In cross-dataset scenarios, performance decreases slightly on UNSW-NB15 but remains strong overall, highlighting that while near-real-time inference is feasible, practical deployment still requires alert calibration to effectively manage alert volume.

3.6. Sensitivity Analysis and Hyperparameter Stability (S4)

To further explore parameter sensitivity analysis in Scenario S4, a limited parameter sweep was conducted on the unsupervised model to measure performance stability across configuration variations. This addition aims to validate that the key findings regarding operational weaknesses remain consistent under varying model configurations, without shifting the study's focus solely toward performance optimization. Sensitivity analysis in Scenario S4 was performed using a light parameter sweep to measure performance volatility under small changes in density- and distance-based models. Testing was conducted on the S0 test data (CICIDS2018 subset), with the following parameter combinations.

Table 11. Unsupervised Model Parameter Grid (Scenario S4)

Model	Parameter	Values Tested	Number of Combinations
DBSCAN	ε (eps)	0.1, 0.5, 1.0	3
	min_samples	5, 10, 20	3
	Total		9 combinations
Isolation Forest	n_estimators	50, 100, 200	3
	contamination	auto, 0.01, 0.05	3
	Total		9 combinations
LOF	n_neighbors	10, 20, 50	3
	threshold (percentile)	90th, 95th, 99th	3
	Total		up to 9 combinations

The results of the sensitivity analysis in Table 11 show that the unsupervised model exhibits significantly different levels of volatility. DBSCAN exhibits very high sensitivity to parameters, with FAR varying from 0.000141 to 0.030663 (approximately $\times$217), while all configurations yield an F1 score of 0. This indicates that although the number of points classified as anomalies is heavily influenced by the parameters ε and MinPts, the model fails to consistently capture relevant attack patterns. Isolation Forest exhibits high parameter stability, with a constant FAR (0.129220) across all parameter combinations; however, performance remains very low (F1 = 0), indicating limited ability to distinguish attack distributions in the normalized feature space. LOF exhibits relatively limited variation, with FAR ranging from 0.149 to 0.151 and F1 between 0.009 and 0.017, suggesting that parameter changes have only a marginal impact on performance.

Table 12. Summary of Unsupervised Model Volatility (Scenario S4)

Family	n	FAR (min)	FAR (max)	Alert Rate (min)	Alert Rate (max)	F1 (min)	F1 (max)
DBSCAN	6	0.000141	0.030663	0.000090	0.019528	0.0	0.0
Isolation Forest	2	0.129220	0.129220	0.082330	0.082330	0.0	0.0
LOF	4	0.149964	0.150629	0.099557	0.100000	0.009	0.017

For comparison, Table 12 shows that hyperparameter optimization was performed exclusively on supervised models (Random Forest and XGBoost) using RandomizedSearchCV with n_iter = 5 and 3-fold cross-validation based on the F1-score. The parameter search space was designed to be moderate in order to maintain a balance between configuration exploration and computational efficiency. This limited optimization strategy was intentionally selected to reflect realistic operational settings where extensive tuning may not always be feasible due to resource and time constraints.

Table 13. Hyperparameter Search Space for Supervised Models

Model	Parameter	Search Value
XGBoost	n_estimators	100, 200
	max_depth	3, 6, 9
	learning_rate	0.01, 0.1, 0.2
	subsample	0.7, 0.9
	colsample_bytree	0.7, 0.9
Random Forest	n_estimators	100, 200
	max_depth	None, 10, 20
	min_samples_split	2, 5
	max_features	sqrt, log2

The optimization results show in Table 13 show that both supervised models continue to achieve near-perfect performance (F1 $\approx$ 1.0) on the source domain (CICIDS2018), even with default or semi-default configurations. This indicates that the dataset's high separability makes the impact of hyperparameter tuning on in-domain performance minimal. This finding reinforces the validity of using lightweight configurations in the S0–S5 evaluation framework. Furthermore, hyperparameter tuning does not alter the study's main conclusion: very high baseline performance does not guarantee operational robustness, particularly against the telemetry degradation observed in scenario S2.

3.7. Fair Comparison and Calibration of Unsupervised Models (S4)

To ensure a fairer comparison between supervised and unsupervised approaches, parameter tuning and calibration were applied to the unsupervised models via a controlled parameter sweep. This approach aims to test whether the observed low performance is due to suboptimal parameter configurations or reflects more fundamental limitations of the unsupervised approach within the feature space used. In this study, unsupervised models (Isolation Forest, LOF, and DBSCAN) demonstrated significantly lower performance (F1 $\approx$ 0.0) compared to supervised models (F1 $\approx$ 1.0). Reviewers questioned whether this comparison was fair, given that unsupervised models rely heavily on threshold calibration and parameter tuning to produce meaningful binary predictions. To address this, parameter adjustments and calibrations were performed via a controlled parameter sweep to ensure that the evaluation was conducted under the fairest possible conditions.

For Isolation Forest, variations in the contamination parameter and the number of estimators were tested to assess sensitivity to assumptions regarding the proportion of anomalies and model complexity. Three contamination values were used: auto (scikit-learn default), 0.01 (low anomaly assumption), and 0.05 (moderate assumption), with the number of trees (n_estimators) set to 50, 100,

and 200. All combinations resulted in F1 = 0.0, indicating that the model's failure did not stem from the threshold or the number of estimators, but rather from the inability to form a meaningful isolation structure in the normalized feature space.

For LOF (Local Outlier Factor), variations in the number of neighbors (n_neighbors = 10, 20, 50) and percentile thresholds (90th, 95th, 99th) were tested to assess sensitivity to the definition of locality and the aggressiveness of detection. The best results were obtained with the configuration k=10 and a 95th percentile threshold, yielding a recall of 0.011 and a false alarm rate of approximately 0.151. These findings indicate that LOF performance is highly sensitive to neighborhood configuration and anomaly threshold selection. In addition, stricter thresholds generally reduced false alarms but also limited the model's ability to detect attack instances. The complete LOF calibration results are presented in the following table.

Table 14. LOF Calibration Results in the Fair Comparison Scenario

Setting	Precision	Recall	F1	FAR
k=10, Th=95	0.040	0.011	0.017	0.151
k=10, Th=99	0.029	0.006	0.009	0.150
k=20, Th=95	0.040	0.011	0.017	0.151
k=20, Th=99	0.029	0.006	0.009	0.150

The results in Table 14 show that changing the parameters has only a marginal effect and does not improve the model's ability to distinguish between benign traffic and attacks. DBSCAN with variations in ε (0.75, 1.0, 1.25) and MinPts (10, 20) still yields an F1 score of 0.0 with a FAR of 0.000141–0.030663, indicating that attack traffic is not separated from noise in the MinMax-normalized feature space due to high distribution overlap. This failure is structural, not merely a calibration issue, due to feature-space overlap, high dimensionality (78 features) that triggers the curse of dimensionality, and statistical similarity between normal traffic and attacks. Thus, the performance difference between supervised and unsupervised methods reflects the fundamental limitations of the unsupervised approach in flow-based IDS, and these findings reinforce the research focus on weakness mapping rather than performance optimization.

3.8. Interpretability Results

We computed SHAP values (TreeExplainer) for the supervised models to interpret dominant drivers behind intrusion classification and to contextualize representative failure cases (FP/FN) [15, 16, 26]. Since our evaluation uses a binary intrusion label (BENIGN vs ATTACK), global feature importance was computed using the mean absolute SHAP values aggregated across samples, as formulated in Equation 2, to obtain stable and analyst-friendly feature rankings. On the CICIDS2018_0214 subset, the global SHAP rankings indicate that model decisions are primarily driven by a combination of (i) port/segment attributes and (ii) rate/window/flag behavior. Across the presented models, the most influential features include fwd_seg_size_min and dst_port, followed by rate-related indicators such as bwd_pkts/s, flow_pkts/s, and fwd_pkts/s, as well as TCP/window and flag features such as init_fwd_win_byts, init_bwd_win_byts, psh_flag_cnt, and ack_flag_cnt. These drivers are consistent with traffic-intensity and session/transport characteristics that differentiate benign versus malicious sessions in flow-based IDS settings. Importantly, many of these features depend on packet counting and timing measurements; therefore, they are also sensitive to telemetry degradation (noise/missingness), which helps explain why supervised performance can drop sharply under S2 stress testing.

To strengthen the weakness mapping perspective beyond global rankings, we additionally present a local SHAP waterfall explanation for a representative false-positive case (Figure 3) [10, 12, 13, 28]. The local explanation shows how specific feature values (e.g., fwd_seg_size_min and dst_port, together with window/flag and rate indicators) push the model toward an "attack" prediction despite the benign ground truth. This supports actionable diagnosis of alert inflation (why certain benign traffic patterns are flagged), and it provides a transparent bridge from model behavior to operational risk interpretation (false alarms vs missed detections) [10, 12, 13, 28].

Global SHAP feature importance for XGBoost (TreeExplainer). Features are ranked by mean absolute SHAP values, aggregated across samples and classes where applicable, indicating each feature's overall contribution to model predictions. Higher SHAP values represent stronger influence on the model decision-making process. This visualization provides insight into the dominant traffic characteristics affecting intrusion detection outcomes and improves the interpretability of the proposed framework.

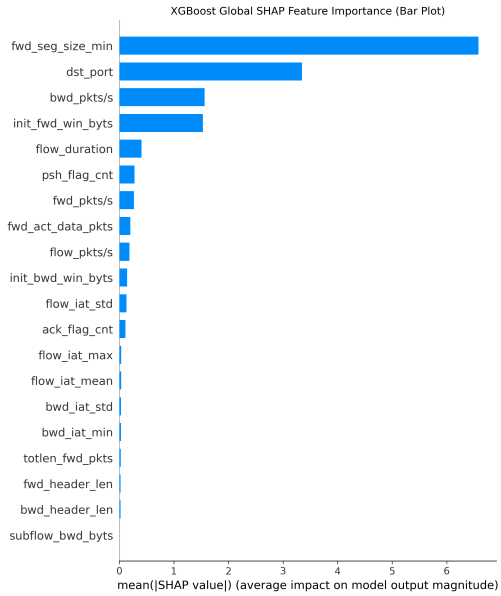


Figure 3. XGBoost Global SHAP Feature Importance

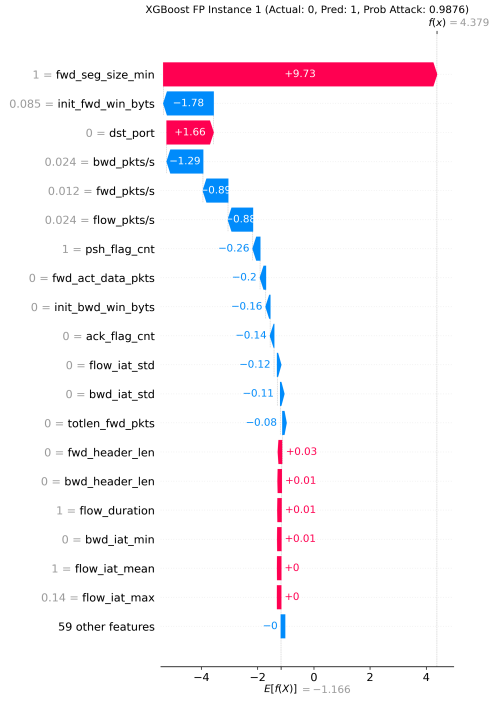


Figure 4. XGBoost Local SHAP Explanation for a Representative False-Positive

As illustrated in Figure 4, the local SHAP explanation for a representative false-positive highlights how specific feature contributions can lead the model to incorrectly classify benign traffic as anomalous. The visualization shows the direction and magnitude of individual feature impacts on the prediction outcome, providing detailed insight into the model's decision process at the instance level. These explanations enable analysts to distinguish between genuine threats and benign but anomalous traffic patterns, supporting risk-informed alert triage and improving operational trust in IDS outputs.

3.9. Discussion

The results show that supervised ensembles (Random Forest and XGBoost) achieve near-perfect baseline ranking on clean benchmark conditions and remain stable under severe train-time imbalance. However, the operational weakness map becomes evident under telemetry degradation (S2): small inference-time Gaussian perturbations can sharply reduce recall/F1, indicating that high benchmark accuracy does not guarantee reliable detection when measurement quality deteriorates in real deployments. Hyperparameter tuning on supervised models such as Random Forest and XGBoost did not alter the study's main findings, as both models already achieved very high performance on the benchmark data due to the inherently high separability of the CICIDS2018 dataset. This indicates that the performance obtained is determined not by specific parameter configurations but by the intrinsic characteristics of the data itself. Conversely, sensitivity analysis of unsupervised models reveals parameter brittleness, a condition in which small changes in parameters lead to significant variations in detection behavior without improving discriminative power. This finding confirms that the primary weakness of unsupervised models lies not in a lack of optimization, but in fundamental instability when dealing with realistic feature distributions.

Unsupervised and density/distance-based baselines exhibit operationally relevant limitations. Isolation Forest and LOF can suffer from low recall and non-trivial false-alarm behavior, while DBSCAN shows strong volatility under small parameter changes, reducing portability and increasing retuning burden. These patterns support the main contribution: scenario-based weakness mapping rather than leaderboard-style benchmarking. However, the comparison between supervised and unsupervised learning methods in this study is not yet fully methodologically balanced. Supervised models inherently benefit from the availability of labels during training, whereas unsupervised methods rely on assumptions about the data's structure and are highly sensitive to parameter selection. In this study, the unsupervised method did not undergo extensive parameter tuning or calibration, so the resulting performance may not reflect its full potential. Thus, the observed performance differences between the two approaches cannot be fully considered an apples-to-apples comparison but rather reflect the practical challenges in applying each method under realistic operational conditions.

The presence of near-perfect PR-AUC and F1 scores also requires careful interpretation. Such results may indicate potential risks of data leakage or overly separable benchmark conditions. Even with appropriate preprocessing and data splitting, benchmark artifacts can artificially inflate performance. The key insight here is that seemingly perfect models can still be operationally fragile. This is evident in the S2 stress scenario, where relatively mild perturbations significantly degrade performance and increase false negative risk. This reinforces the need to avoid overestimating deployment readiness based solely on benchmark metrics.

From an operational perspective, usability extends beyond computational efficiency. The observed alert rate of approximately 0.36 suggests that more than one-third of traffic would be flagged under default decision thresholds, which is impractical for Security Operations Center workflows without additional mechanisms such as alert aggregation, suppression, or cost-sensitive thresholding. Therefore, claims of near-real-time capability must be accompanied by appropriate alert-calibration strategies to ensure that detection outputs remain actionable and manageable.

Interpretability further plays a critical role in translating model weaknesses into operational risk. Techniques such as SHAP provide a transparent bridge between model behavior and operational consequences by identifying dominant traffic features and enabling targeted inspection of false-positive and false-negative cases [12, 13]. Within the weakness-mapping framework, interpretability is not merely an auxiliary component but a key mechanism for diagnosing why errors occur, improving analyst trust, and supporting risk-informed decision-making.

Several limitations should also be acknowledged. In terms of internal validity, near-perfect baseline performance necessitates scrutiny regarding potential data leakage. This study mitigates such risks by fitting preprocessing steps exclusively to the training data and applying perturbations only to the test data. From a construct validity perspective, the use of Gaussian noise represents a simplified approximation of telemetry degradation and may not capture more structured or systematic data issues. Additionally, excessive noise can obscure other effects, meaning the S2 scenario should be interpreted as a stress boundary rather than a precise real-world simulation. It should also be emphasized that the comparison between supervised and unsupervised models has inherent limitations, as unsupervised methods do not use labeled data, whereas supervised models explicitly optimize class separation. Therefore, performance comparisons should be interpreted with caution and understood as differences in operational behavior across distinct detection paradigms rather than as fully equivalent evaluations.

External validity is also constrained by practical considerations. Full scenario evaluations may be limited to computationally feasible datasets, which can restrict the completeness of cross-dataset analysis. As such, generalization of results should be viewed as indicative rather than universally representative. Cross-dataset analysis also revealed significant domain shifts between CICIDS2018 and UNSW-NB15, characterized by distribution mismatches and the invalidation of learned decision boundaries. These findings indicate that strong in-domain performance does not necessarily translate into robustness across different environments, reinforcing the need to interpret benchmark accuracy with caution in real-world deployment contexts.

Overall, these findings point toward several important directions for future research, including the development of robustness-oriented preprocessing and feature representations to reduce sensitivity to telemetry degradation, the integration of drift-aware validation to detect silent performance decay, and the implementation of alert-policy calibration strategies to ensure that detection systems remain stable, interpretable, and operationally feasible.

4. CONCLUSION

This study presents a directed replication of representative machine learning–based intrusion and anomaly detection baselines and develops a scenario-based operational weakness map under realistic stress conditions (S0–S5). The findings show that supervised ensemble models such as Random Forest and XGBoost achieve excellent benchmark performance and remain robust under severe train-time imbalance. However, their primary limitation emerges under telemetry degradation (S2), where even small inference-time perturbations can lead to significant drops in recall and F1, increasing the risk of missed attacks. In contrast, unsupervised and density/distance-based methods reveal additional operational challenges, including higher false-alarm rates, limited recall, and strong sensitivity to parameter configurations, which reduce stability and portability across deployment environments.

Although near-real-time micro-batch inference is computationally feasible, practical adoption in Security Operations Center environments is constrained by the high alert burden (alert_rate ≈ 0.36), underscoring the need for effective calibration and alert management strategies to prevent analyst fatigue. Overall, the study highlights that the reliability of intrusion detection systems depends not only on accuracy but also on robustness to data degradation, interpretability to support analyst decision-making, and the ability to control alert volume. These findings motivate future research on noise-robust feature representations, drift-aware validation, and cost-sensitive alert calibration to improve system resilience and ensure operational feasibility in high-volume and degraded telemetry scenarios.

5. ACKNOWLEDGEMENTS

The authors would like to express their gratitude to the research and data science community for providing publicly accessible cybersecurity datasets, including CICIDS2017, CICIDS2018, UNSW-NB15, and RanSMAP, which made this study possible. The authors also acknowledge the academic and technical support provided by the Informatics Doctoral Program and the Cybersecurity Research Group during the experimental design and validation stages.

6. DECLARATIONS

AI USAGE STATEMENT

During the preparation of this work, the author(s) used ChatGPT (OpenAI), Gemini, and Grammarly to support language editing, improve clarity and coherence of academic writing, and assist in organizing the manuscript structure (e.g., drafting section transitions and refining technical explanations). After using this tool, the author(s) reviewed and edited the content as needed and take(s) full responsibility for the accuracy, originality, and integrity of the publication's content.

AUTHOR CONTIBUTION

All authors contributed to the study conception and design. The first author conducted the experiments, implemented the benchmarking and stress-testing scenarios, and performed the data analysis. The author(s) interpreted the results, drafted the manuscript, and revised it critically for important intellectual content. All author(s) read and approved the final manuscript.

FUNDING STATEMENT

This research received no external funding.

COMPETING INTEREST

The author(s) declare that they have no competing interests.

REFERENCES

[1] E. Krzysztoń, I. Rojek, and D. Mikołajewski, "A Comparative Analysis of Anomaly Detection Methods in IoT Networks: An Experimental Study," *Applied Sciences*, vol. 14, no. 24, p. 11545, Dec. 2024, https://doi.org/10.3390/app142411545.

[2] S. Narmadha and N. Balaji, "Improved network anomaly detection system using optimized autoencoder - LSTM," *Expert Systems with Applications*, vol. 273, p. 126854, May 2025, https://doi.org/10.1016/j.eswa.2025.126854.

[3] F. Mahardika, E. Utami, Kusrini, and F. W. Wibowo, "Towards Transparent Cyber Threat Detection : A Systematic Literature Review on the Role of Explainable AI (XAI) in Information Security Risk Management (2018-2025)," in *2025 13th International Conference on Cyber and IT Service Management (CITSM)*. Jakarta, Indonesia: IEEE, Sep. 2025, pp. 1–4, https://doi.org/10.1109/CITSM67730.2025.11291277.

[4] ——, "A Systematic Literature Review on Machine Learning-Based Information Security Risk Management for Higher Education Institutions," in *2025 IEEE International Conference on Advanced Information Scientific Development (ICAISD)*. Jakarta, Indonesia: IEEE, Nov. 2025, pp. 84–89, https://doi.org/10.1109/ICAISD68166.2025.11385757.

[5] A. H. Muhammad, A. Nasiri, and A. Harimurti, "Machine learning methods for classification and prediction information security risk assessment," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 14, no. 1, pp. 457–465, Feb. 2025, https://doi.org/10.11591/ijai.v14.i1.pp457-465.

[6] Y. Xin, L. Kong, Z. Liu, Y. Chen, Y. Li, H. Zhu, M. Gao, H. Hou, and C. Wang, "Machine Learning and Deep Learning Methods for Cybersecurity," *IEEE Access*, vol. 6, pp. 35 365–35 381, 2018, https://doi.org/10.1109/ACCESS.2018.2836950.

[7] N. G. Pardeshi and D. V. Patil, "A two-layered collaborative approach for network intrusion detection system using blended shallow learning gaussian naïve bayes and support vector machine models," *International Journal of Advances in Intelligent Informatics*, vol. 11, no. 3, pp. 459–479, Aug. 2025, https://doi.org/10.26555/ijain.v11i3.2035.

[8] Y. Almutairi, B. Alhazmi, and A. Munshi, "Network Intrusion Detection Using Machine Learning Techniques," *Advances in Science and Technology Research Journal*, vol. 16, no. 3, pp. 193–206, Jul. 2022, https://doi.org/10.12913/22998624/149934.

[9] D. G. Hakke, "Performance Evaluation of Machine Learning-Based Intrusion Detection Using NSL-KDD, UNSW-NB15 and CICIDS2017 Datasets," *International Journal of Applied Mathematics*, vol. 38, no. 3s, pp. 447–469, Sep. 2025, https://doi.org/10.12732/ijam.v38i3s.160.

[10] S. A. Ajagbe, J. B. Awotunde, and H. Florez, "Intrusion Detection: A Comparison Study of Machine Learning Models Using Unbalanced Dataset," *SN Computer Science*, vol. 5, no. 8, p. 1028, Nov. 2024, https://doi.org/10.1007/s42979-024-03369-0.

[11] D. Paolini, P. Dini, A. Elhanashi, and S. Saponara, "Advanced Fault Detection and Diagnosis Exploiting Machine Learning and Artificial Intelligence for Engineering Applications," *Electronics*, vol. 15, no. 2, p. 476, Jan. 2026, https://doi.org/10.3390/electronics15020476.

[12] V. Z. Mohale and I. C. Obagbuwa, "Evaluating machine learning-based intrusion detection systems with explainable AI: Enhancing transparency and interpretability," *Frontiers in Computer Science*, vol. 7, p. 1520741, May 2025, https://doi.org/10.3389/fcomp.2025.1520741.

[13] L. Bernal, G. Rastelli, and L. Pinzi, "Improving Machine Learning Classification Predictions through SHAP and Features Analysis Interpretation," *Journal of Chemical Information and Modeling*, vol. 65, no. 21, pp. 11 716–11 732, Nov. 2025, https://doi.org/10.1021/acs.jcim.5c02015.

[14] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, Feb. 2020, https://doi.org/10.1016/j.jisa.2019.102419.

[15] G. R. Ginni and S. L. Chakravarthy, "A Hybrid Framework for Robust Anomaly Detection: Integrating Unsupervised and Supervised Learning with Advanced Feature Engineering," *International Journal of Computational and Experimental Science and Engineering*, vol. 11, no. 2, pp. 1993–2017, Apr. 2025, https://doi.org/10.22399/ijcesen.1383.

[16] P. Lavanya, R. P. Singh, U. Kumaran, and P. Kumar, "Gradient Boosting classifier performance evaluation using Generative Adversarial Networks," *Procedia Computer Science*, vol. 235, pp. 3016–3024, 2024, https://doi.org/10.1016/j.procs.2024.04.285.

[17] F. Ebrahimi, R. Javidan, R. Akbari, and Y. Hosseini, "Intrusion detection in the internet of things using convolutional neural networks: An explainable AI approach," *Cybersecurity*, vol. 8, no. 1, p. 66, Sep. 2025, https://doi.org/10.1186/s42400-025-00369-2.

[18] V. Kumar, S. S. Saheb, Preeti, A. Ghayas, S. Kumari, J. K. Chandel, S. K. Pandey, and S. Kumar, "AI-Based Hybrid Models for Predicting Loan Risk in the Banking Sector," *Big Data Mining and Analytics*, vol. 6, no. 4, pp. 478–490, Dec. 2023, https://doi.org/10.26599/BDMA.2022.9020037.

[19] H. Zhang, Z. Xiao, J. Gu, and Y. Liu, "A network anomaly detection algorithm based on semi-supervised learning and adaptive multiclass balancing," *The Journal of Supercomputing*, vol. 79, no. 18, pp. 20 445–20 480, Dec. 2023, https://doi.org/10.1007/s11227-023-05474-y.

[20] M. Aamir and S. M. Ali Zaidi, "Clustering based semi-supervised machine learning for DDoS attack classification," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 4, pp. 436–446, May 2021, https://doi.org/10.1016/j.jksuci.2019.02.003.

[21] M. Hirano and R. Kobayashi, "RanSMAP: Open dataset of Ransomware Storage and Memory Access Patterns for creating deep learning based ransomware detectors," *Computers & Security*, vol. 150, p. 104202, Mar. 2025, https://doi.org/10.1016/j.cose.2024.104202.

[22] Hari Vinayak M.V. and Jarin T., "A hybrid model for detecting intrusions using stacked autoencoders and extreme gradient boosting," *Computers & Security*, vol. 150, p. 104212, Mar. 2025, https://doi.org/10.1016/j.cose.2024.104212.

[23] N. Amroune, M. Benazi, and L. Sayad, "An Adaptative Eps Parameter of DBSCAN Algorithm for Identifying Clusters with Heterogeneous Density," *Computación y Sistemas*, vol. 28, no. 2, Jun. 2024, https://doi.org/10.13053/cys-28-2-4600.

[24] J. I. Iturbe-Araya and H. Rifà-Pous, "Enhancing unsupervised anomaly-based cyberattacks detection in smart homes through hyperparameter optimization," *International Journal of Information Security*, vol. 24, no. 1, p. 45, Feb. 2025, https://doi.org/10.1007/s10207-024-00961-6.

[25] O. Alghushairy, R. Alsini, T. Soule, and X. Ma, "A Review of Local Outlier Factor Algorithms for Outlier Detection in Big Data Streams," *Big Data and Cognitive Computing*, vol. 5, no. 1, pp. 1–24, Mar. 2021, https://doi.org/10.3390/bdcc5010001.

[26] S. T. Hamidou and A. Mehdi, "Enhancing IDS performance through a comparative analysis of Random Forest, XGBoost, and Deep Neural Networks," *Machine Learning with Applications*, vol. 22, p. 100738, Dec. 2025, https://doi.org/10.1016/j.mlwa.2025.100738.

[27] S. Oh, S. Sohn, C. Kim, and M. Park, "MCH-Ensemble: Minority Class Highlighting Ensemble Method for Class Imbalance in Network Intrusion Detection," *Applied Sciences*, vol. 15, no. 23, p. 12647, Nov. 2025, https://doi.org/10.3390/app152312647.

[28] R. Mohite and L. Ouarbya, "Interpretable Anomaly Detection: A Hybrid Approach Using Rule-Based and Machine Learning Techniques," in *2024 IEEE 9th International Conference for Convergence in Technology (I2CT)*. Pune, India: IEEE, Apr. 2024, pp. 1–10, https://doi.org/10.1109/I2CT61223.2024.10543396.