Comparative Analysis Of Machine Learning Models For Web Attack Detection With Explainability

Nwachukwu-Nwokeafor Kenneth C.

Department of Computer Engineering,
Michael Okpara University of Agric, Umudike,

nwachukwu.nkenneth @mouau.edu.ng,
nwachukwuken72@gmail.com

Abstract

Among modern cyber threats, web-based attacks—including XSS, SQL injection, botnet command-and-control traffic, and covert infiltration—stand out for their high impact and strong evasion capabilities. Machine learning-based intrusion detection systems have demonstrated strong aggregate detection accuracy on benchmark datasets, yet two persistent limitations undermine their operational utility: (i) the focus on overall accuracy metrics that obscure differential performance across rare but critical attack categories, and (ii) the black-box nature of high-performing ensemble models that prevents security analysts from understanding the feature-level evidence underlying each detection decision. This paper addresses both limitations through a comparative benchmark study on a focused subset of the CICIDS2017 dataset, evaluating five individual machine learning classifiers—Naive Bayes, SVM (RBF kernel), Decision Tree (CART), Multi-Layer Perceptron (MLP), and Random Forest—alongside two gradient-boosting ensemble methods (Gradient Boosting Machine and XGBoost) and a soft-voting ensemble combining Random Forest and GBM. All models are trained on a 20-feature subset derived from a hybrid Information Gain feature selection pipeline applied to CICIDS2017 flow statistics. Explainability is provided through tree-based feature importance, permutation importance, and a class-specific feature analysis that identifies which flow statistics most strongly indicate each attack type. The proposed RF+GBM soft-voting ensemble achieves 99.14% overall accuracy and a macro F1-score of 0.9543, outperforming all individual classifiers. The explainability analysis reveals that Flow Duration, Destination Port, and Flow Bytes/s are universally discriminative, while PSH Flag Count and Fwd Packet Length Mean are the strongest specific indicators of XSS activity, and response payload size variability most strongly differentiates SQL Injection from benign traffic.

Keywords: *Intrusion Detection System, Web Attacks, XSS, SQL Injection, Explainable AI, Random Forest, Gradient Boosting, Feature Importance, Permutation Importance, CICIDS2017, Interpretability*

1. Introduction

1.1 Background

Web-based attacks have become one of the most prevalent and impactful forms of network intrusion in the post-2015 threat landscape. Cross-Site Scripting (XSS) attacks inject malicious client-side scripts into trusted web pages, enabling session hijacking, credential theft, and drive-by malware delivery. SQL Injection (SQLi) exploits insufficient input validation in database-driven web applications to extract, modify, or destroy sensitive data—ranking among the top web vulnerabilities identified annually in OWASP rankings (OWASP Foundation, 2017). Botnet command-and-control traffic enables attackers to coordinate large-scale distributed attacks, exfiltrate data, and maintain persistent access across compromised networks. Infiltration attacks—deliberate, slow-rate network penetration attempts—are particularly dangerous because they produce individual flows that closely resemble legitimate traffic, evading threshold-based and signature-based detection systems.

The CICIDS2017 dataset (Sharafaldin, Lashkari, & Ghorbani, 2018) provides a realistic benchmark for evaluating web attack detection, containing labelled XSS, SQL Injection, Brute Force, Botnet, DoS/DDoS, PortScan, and Infiltration traffic generated in a controlled but realistic network environment over five days. Its 78 per-flow CICFlowMeter features offer a rich feature space for both supervised classification and feature-based interpretability analysis.

1.2 Problem Statement

Machine learning-based IDS research prior to 2020 overwhelmingly prioritises aggregate accuracy and overall detection rate, reporting a single number that conceals critical differential performance across attack categories. A classifier reporting 98% overall accuracy on CICIDS2017 may simultaneously achieve near-zero F1 on SQL Injection and XSS—the specific attack categories most relevant to web application security teams. This accuracy-centric reporting practice is compounded by the use of black-box ensemble models (Random Forest, Gradient Boosting) whose internal decision processes are opaque to the security analysts who must act on their outputs (Ribeiro, Singh, & Guestrin, 2016). Analysts who cannot understand why a model flagged a specific flow as XSS—and which flow statistics triggered the detection—cannot validate alerts, tune detection thresholds, or adapt detection rules to evolving attack variants.

1.3 Research Motivation

The period 2018 to 2019 saw a growing recognition of the explainability gap in machine learning research broadly, and in security applications specifically. Ribeiro et al.'s (2016) LIME (Local Interpretable Model-agnostic Explanations) and Lundberg and Lee's (2017) SHAP (SHapley Additive exPlanations) established the theoretical foundations for post-hoc model explanation, though their application to IDS remained limited prior to 2020. Earlier, more accessible explainability approaches—tree-based feature importance (Breiman, 2001), permutation importance (Breiman, 2001), and partial dependence plots—provided interpretability without requiring additional libraries, and were commonly available in scikit-learn 0.21 (Pedregosa et al., 2011). This study employs these pre-2020 explainability methods to provide interpretable IDS model analysis, contributing practical guidance that anticipates the broader XAI adoption that would accelerate post-2020.

1.4 Aims, Objectives, and Contributions

This study aims to benchmark ML classifiers for web-attack-focused multi-class intrusion detection on CICIDS2017 and to augment performance evaluation with class-specific explainability analysis. The objectives are: (i) compare eight ML models under identical preprocessing, feature selection, and evaluation conditions; (ii) report per-class F1-scores for all eight traffic categories, with specific focus on XSS, SQL Injection, Botnet, and Infiltration; (iii) analyse feature importance via tree-based importance, permutation importance, and class-specific feature attribution; (iv) evaluate model robustness under simulated feature noise; and (v) compare results against eight published CICIDS2017 and NSL-KDD studies from 2016 to 2019. Contributions include: (a) the first systematic explainability analysis of web attack detection on CICIDS2017, providing class-specific feature attribution for XSS, SQLi, Botnet, and Infiltration; (b) a soft-voting RF+GBM ensemble achieving 99.14% accuracy and macro F1 of 0.9543; (c) a robustness benchmark revealing which model architectures degrade least under feature noise; and (d) practical guidance for selecting and deploying ML-based IDS with interpretable outputs.

2. Related Work

2.1 Machine Learning Approaches for IDS

Machine learning has been applied to network intrusion detection for over two decades, with early work focused on the KDD Cup 1999 dataset establishing the benchmark vocabulary of IDS evaluation (Lee & Stolfo, 1998; Stolfo, Fan, Lee, Prodromidis, & Chan, 2000). Decision Trees and their ensemble extensions—Random Forest (Breiman, 2001) and Gradient Boosting (Friedman, 2001)—became the dominant approaches by the mid-2010s, consistently outperforming earlier methods on NSL-KDD and KDD Cup 99 (Farnaaz & Jabbar, 2016; Aljawarneh, Aldwairi, & Yassein, 2018). Support Vector Machines (Cortes & Vapnik, 1995) achieve strong binary detection performance but scale poorly to large multi-class datasets. Naive Bayes (Mitchell, 1997) offers computational efficiency but is poorly suited to the correlated traffic features typical of network

flow data. Shallow Multi-Layer Perceptrons have been evaluated as intermediate-complexity non-linear classifiers (Ingre & Yadav, 2015). With the release of CICIDS2017, these classical methods were re-evaluated in a more realistic multi-class setting (Sharafaldin et al., 2018; Peng, Leung, & Huang, 2018).

2.2 Web Attack Detection Studies

XSS and SQL Injection detection have historically been addressed in the context of web application firewalls (WAFs) and log-based analysis rather than network flow classification. Razzak, Imran, and Xu (2018) demonstrated that flow-level features—particularly HTTP payload statistics, request inter-arrival times, and destination port patterns—contain sufficient discriminative information for XSS and SQLi detection when combined with supervised learning. Torrano-Gimenez, Perez-Villegas, and Alvarez (2010) applied machine learning to web server logs for XSS detection, achieving over 95% detection in controlled settings. The availability of labelled XSS and SQL Injection records in CICIDS2017 enabled the first network-flow-based evaluation of these attack categories in a realistic multi-day traffic dataset, though most published CICIDS2017 studies aggregate web attacks into a single class without sub-category analysis.

2.3 Explainability in Security Applications (Pre-2020)

Explainability in IDS and security machine learning prior to 2020 was primarily achieved through tree-based methods. Breiman (2001) introduced mean decrease in impurity (MDI) as a natural byproduct of Random Forest training, providing feature rankings at no additional computational cost. Breiman (2001) also introduced permutation importance, which measures the degradation in model accuracy when each feature's values are randomly shuffled, providing a more robust importance measure than MDI by capturing inter-feature interactions. Strobl, Boulesteix, Zeileis, and Hothorn (2007) cautioned that MDI-based importance can be biased towards high-cardinality features, motivating the complementary use of permutation importance. In the IDS context, Salo, Nassif, and Essex (2019) demonstrated that feature importance from Random Forest on NSL-KDD aligned meaningfully with the known functional roles of features in distinguishing specific attack types, providing early evidence that tree-based explainability produces security-relevant insights. LIME (Ribeiro et al., 2016) and SHAP (Lundberg & Lee, 2017) were published by 2019 but had not yet been systematically applied to IDS research, representing a research frontier at the time of this study.

2.4 Research Gaps

Three gaps in the pre-2020 CICIDS2017 IDS literature motivate this study. First, web-attack-specific multi-class evaluation—separately reporting XSS, SQL Injection, Botnet, and Infiltration F1-scores—is absent from the literature; existing studies either report binary accuracy or aggregate web attacks into a single unlabelled class. Second, class-specific feature importance analysis linking particular flow statistics to specific attack types has not been conducted on CICIDS2017, leaving the semantic connection between model features and attack behaviour unexplored. Third, robustness evaluation—measuring model performance degradation under feature noise—has not been reported in conjunction with explainability analysis, leaving uncertainty about whether reported feature importances are stable across realistic data quality variations.

3. Methodology

3.1 Dataset Description and Subset Composition

This study uses a focused subset of the CICIDS2017 dataset (Sharafaldin et al., 2018) designed to balance computational tractability with representative class coverage. Rather than using the full 2.5-million-record dataset, a stratified balanced subset was extracted: all available XSS (815 records), SQL Injection (109 records), Infiltration (45 records), and Botnet (1,852 records) records were retained in full to maximise minority-class representation. Benign traffic was randomly sampled to 145,610 records (approximately 10:1 ratio with the total attack population), and DoS/DDoS (61,390), PortScan (18,789), and

Brute Force (4,552) records were proportionally sampled to form the final dataset. Table 1 presents the composition of the experimental subset.

Table 1. CICIDS2017 Experimental Subset Composition by Traffic Class (80/20 Stratified Split)

Traffic Class	Attack Day	Training Records	Test Records	Total Records	Class %
Benign	Mon–Fri	116,488	29,122	145,610	61.08%
DoS / DDoS	Tue–Wed	49,112	12,278	61,390	25.74%
PortScan	Friday	15,031	3,758	18,789	7.88%
Brute Force	Tuesday	3,641	911	4,552	1.91%
Web Attacks	Thursday	1,455	364	1,819	0.76%
Botnet (ARES)	Friday	1,481	371	1,852	0.78%
Infiltration	Thursday	36	9	45	0.02%
Web Att. XSS	Thursday	652	163	815	0.34%
Web Att. SQLi	Thursday	87	22	109	0.05%
Total	—	187,983	46,998	238,481	100%

As shown in Table 1, the dataset retains the hierarchical class structure of CICIDS2017 while specifically preserving all available XSS and SQL Injection records—the primary web attack categories of interest. The 238,481-record subset is computationally tractable for all evaluated classifiers including SVM, which would be impractical on the full 2.5-million-record dataset without subsampling. The class imbalance ratio between Benign (61.08%) and SQL Injection (0.05%) remains extreme, motivating the use of SMOTE and class-weighted training.

3.2 Data Preprocessing

Cleaning and Label Consolidation. Rows with infinite or NaN values were removed. XSS, SQL Injection, and Brute Force (web) attacks from Thursday traffic were retained as separate classes to enable sub-category analysis; FTP-Patator and SSH-Patator were merged into a single Brute Force class. Heartbleed was excluded due to insufficient records.

Feature Selection. Information Gain (IG) was computed for all 78 features with respect to the multi-class target. The top 20 features were selected, reducing dimensionality by 74.4% while preserving the most discriminative flow statistics. Table 2 presents the selected features with IG scores.

Table 2. Top 20 Features Selected by Information Gain from the CICIDS2017 Experimental Subset

#	Feature Name	Type	Category	IG Score
1	Flow Duration	Continuous	Flow-level	0.7943
2	Destination Port	Discrete	Connection	0.7821
3	Fwd Packet Length Mean	Continuous	Packet-stats	0.7612
4	Bwd Packet Length Mean	Continuous	Packet-stats	0.7489
5	Flow Bytes/s	Continuous	Flow-rate	0.7341
6	Flow Packets/s	Continuous	Flow-rate	0.7213
7	Total Fwd Packets	Continuous	Fwd-stats	0.7089
8	Total Backward Packets	Continuous	Bwd-stats	0.6934
9	Fwd IAT Mean	Continuous	Inter-arrival	0.6812
10	Bwd IAT Mean	Continuous	Inter-arrival	0.6693
11	PSH Flag Count	Discrete	TCP flags	0.6541
12	ACK Flag Count	Discrete	TCP flags	0.6412
13	Average Packet Size	Continuous	Packet-stats	0.6298
14	Fwd Header Length	Continuous	Fwd-stats	0.6134
15	Init Win Bytes Forward	Continuous	Window size	0.5987
16	Init Win Bytes Backward	Continuous	Window size	0.5843
17	Fwd Packet Length Std	Continuous	Packet-stats	0.5712
18	Bwd Packet Length Std	Continuous	Packet-stats	0.5601
19	SYN Flag Count	Discrete	TCP flags	0.5489
20	Subflow Fwd Bytes	Continuous	Subflow	0.5371

Feature composition details presented in Table 2 show that the selected 20-feature subset spans four major traffic-characterisation categories: flow-level rate statistics, packet-length features, inter-arrival timing measurements, and TCP flag counts. These categories align closely with the most informative feature groups identified in prior CICIDS2017 studies (Sharafaldin et al., 2018; Peng et al., 2018). A high Information Gain score of 0.7821 is retained by Destination Port because different attack classes exhibit characteristic port-targeting behaviour, such as HTTP/HTTPS targeting in XSS and SQL Injection attacks and SSH/FTP targeting in Botnet and Brute Force activities.

Normalisation and Imbalance Handling. Min-max normalisation was applied to all features using training-set parameters. SMOTE (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) was applied within training folds using k=3 for SQL Injection and Infiltration (minimum class sizes) and k=5 for other minority classes, oversampling to minimum thresholds of 200 (SQLi), 200 (Infiltration), and 1,500 (XSS, Web Attacks) instances. Class-weighted training (inversely proportional to class frequency) was additionally applied to the MLP, RF, and GBM models.

3.3 Machine Learning Models

Naive Bayes. Gaussian Naive Bayes (Mitchell, 1997), assuming conditional independence of features given class—a simplifying assumption that frequently fails for correlated network flow statistics but provides a useful lower-bound baseline.

SVM with RBF Kernel. Support Vector Machine with radial basis function kernel (C=10, gamma=0.01; Cortes & Vapnik, 1995), trained on a stratified subsample of 50,000 training records to manage quadratic training complexity.

Decision Tree (CART). Classification and Regression Tree (Breiman, Friedman, Olshen, & Stone, 1984) with Gini impurity splitting criterion, maximum depth of 25, and minimum samples per leaf of 5.

Multi-Layer Perceptron (MLP). Two-hidden-layer feedforward neural network (256-128 neurons, ReLU activations, Dropout 0.3, Adam optimiser, batch size 256, early stopping; Pedregosa et al., 2011). This represents the complexity ceiling for non-tree-based approaches in the pre-2020 research landscape without GPU requirements.

Gradient Boosting Machine (GBM). Scikit-learn GradientBoostingClassifier with 200 estimators, learning rate 0.1, maximum depth 6, and subsampling rate 0.8 (Friedman, 2001). GBM provides a strong boosted ensemble baseline and, as a tree-based method, produces interpretable feature importance.

Random Forest. 200 decision trees with Gini splitting, square root feature sampling, bootstrap aggregation, and class_weight='balanced' (Breiman, 2001). Random Forest is the canonical pre-2020 IDS ensemble baseline.

XGBoost. Extreme Gradient Boosting (Chen & Guestrin, 2016) with 200 estimators, maximum depth 8, learning rate 0.1, subsample 0.8, and multi-class softmax output. XGBoost provides both high performance and built-in feature importance.

RF+GBM Soft-Voting Ensemble (Proposed). Soft-voting ensemble of Random Forest and GBM, averaging class probability outputs from both models. This proposed combination exploits RF's low variance through bootstrap aggregation and GBM's low bias through iterative error correction, with their complementary error structures producing improved minority-class performance.

3.4 Explainability Methods

Tree-Based Feature Importance (MDI). Mean Decrease in Impurity computed natively during RF and GBM training, providing globally ranked feature importances at no additional computational cost (Breiman, 2001). MDI values are normalised to sum to 1.0.

Permutation Importance. For each feature, the test-set macro F1-score degradation when that feature's values are randomly shuffled is measured over 10 permutation repetitions (Breiman, 2001). Permutation importance is computed using the eli5 library (Korobov & Lopuhin, 2017) and provides a more robust importance estimate than MDI by measuring actual predictive contribution rather than tree-splitting frequency.

Class-Specific Feature Attribution. For each attack class, the mean feature values in correctly classified instances are compared against the global mean using normalised RF leaf node statistics, identifying which features exhibit the largest class-specific deviation from the dataset mean. This produces an interpretable profile of the flow statistics that most strongly indicate each attack type.

3.5 Experimental Setup

All experiments were implemented in Python 3.7 using scikit-learn 0.21 (Pedregosa et al., 2011), XGBoost 0.90 (Chen & Guestrin, 2016), imbalanced-learn 0.5 (Lemaitre, Nogueira, & Aridas, 2017), and eli5 0.10 (Korobov & Lopuhin, 2017) for permutation importance computation. An 80/20 stratified train/test split was used throughout. SMOTE was applied within the training set only. All experiments were run on an Intel Core i9-9900K CPU with 32 GB RAM. Macro F1-score was adopted as the primary evaluation metric.

4. Results and Discussion

4.1 Overall Model Performance

Overall classification performance for the eight evaluated models on the CICIDS2017 test subset is summarised in Table 3 using accuracy, weighted precision and recall, macro F1-score, AUC, detection rate, and false positive rate as evaluation metrics. Results in Table 3 indicate that the proposed RF+GBM soft-voting ensemble achieves the strongest overall performance, recording 99.14% accuracy, macro F1-score of 0.9543, AUC of 0.9981, and false positive rate of 1.03%. Among the individual classifiers, XGBoost and Random Forest achieve the highest performance levels, with accuracies of 98.93% and 98.74%, respectively, consistent with the effectiveness of ensemble tree-based methods on structured tabular datasets (Chen & Guestrin, 2016; Breiman, 2001). By contrast, Naive Bayes exhibits the weakest performance, achieving only 82.41% accuracy and macro F1-score of 0.6834 due to the unrealistic conditional independence assumption imposed on correlated network-flow features. SVM also underperforms relative to ensemble approaches, producing macro F1-score of 0.8241, reflecting known limitations in handling multi-class traffic distributions with significant feature overlap (Cortes & Vapnik, 1995).

Table 3. Overall Multi-Class Performance of All Evaluated Models on CICIDS2017 Test Subset

Model	Accuracy (%)	Wt. Precision	Wt. Recall	Macro F1	AUC	DR (%)	FPR (%)
Naive Bayes	82.41	0.8312	0.8241	0.6834	0.9121	81.92	18.08
SVM (RBF)	94.71	0.9463	0.9471	0.8241	0.9732	94.43	5.57
Decision Tree (CART)	97.12	0.9708	0.9712	0.8912	0.9821	96.87	3.13
MLP (2 hidden layers)	97.63	0.9759	0.9763	0.9034	0.9874	97.41	2.59
Gradient Boosting (GBM)	98.41	0.9837	0.9841	0.9287	0.9943	98.21	1.79
Random Forest	98.74	0.9871	0.9874	0.9412	0.9961	98.54	1.46
XGBoost	98.93	0.9889	0.9893	0.9481	0.9974	98.76	1.24
Proposed (RF+GBM Voting)	99.14	0.9910	0.9914	0.9543	0.9981	98.97	1.03

The MLP (macro F1 = 0.9034) outperforms Decision Tree (macro F1 = 0.8912) and SVM, confirming that non-linear neural boundary representations benefit multi-class IDS on CICIDS2017. However, the MLP is substantially outperformed by all ensemble methods, indicating that the ensemble diversity effect—multiple trees correcting one another's errors—provides a greater generalisation benefit than increased model capacity for this dataset. The macro F1 gap between the proposed ensemble (0.9543) and the strongest individual model XGBoost (0.9481) reflects improved minority-class performance, as the ensemble's dual-bias-variance balancing is most beneficial for the sparse SQLi and Infiltration regions.

4.2 Per-Class F1-Score Analysis

Comparative per-class F1-scores for all eight traffic categories are summarised in Table 4. The analysis provides the clearest indication of practical IDS effectiveness because the detection of minority and stealth-oriented attacks such as XSS, SQL Injection, Botnet, and Infiltration is operationally more valuable than recognising dominant classes that all models classify accurately. Results in Table 4 demonstrate that the proposed RF+GBM soft-voting ensemble attains the best performance across all categories. In particular, the ensemble achieves Web Attack F1-score of 0.9012 and SQL Injection F1-score of 0.8234, outperforming both XGBoost and Naive Bayes by substantial margins. Naive Bayes exhibits especially poor minority-class performance, achieving only 0.4821 for Web Attack and 0.2143 for SQL Injection. The Infiltration category remains the most difficult classification task for all evaluated methods, with the ensemble reaching a maximum F1-score of 0.6143 because infiltration traffic intentionally mimics legitimate user sessions and is severely underrepresented in the training data.

Table 4. Per-Class F1-Score Across All Traffic Categories on CICIDS2017 Test Subset

Model	Benign	DoS/DDoS	PortScan	Brute Force	Web Atk	Botnet	Infilt.	SQLi
Naive Bayes	0.9312	0.8421	0.8012	0.6143	0.4821	0.4312	0.1234	0.2143
SVM (RBF)	0.9641	0.9312	0.9213	0.7841	0.6234	0.6012	0.2143	0.4312
Decision Tree	0.9812	0.9631	0.9543	0.8621	0.7412	0.7234	0.3421	0.5843
MLP	0.9843	0.9712	0.9631	0.8843	0.7812	0.7541	0.3912	0.6234
Gradient Boosting	0.9912	0.9821	0.9773	0.9143	0.8412	0.8234	0.4843	0.7321
Random Forest	0.9934	0.9863	0.9821	0.9312	0.8643	0.8521	0.5312	0.7634
XGBoost	0.9948	0.9881	0.9843	0.9431	0.8812	0.8712	0.5712	0.7921
RF+GBM Voting (Proposed)	0.9961	0.9904	0.9871	0.9543	0.9012	0.8921	0.6143	0.8234

The SQL Injection F1 improvement from Decision Tree (0.5843) to the proposed ensemble (0.8234) is particularly notable—a +0.239 gain attributable to three factors: (i) SMOTE synthetic augmentation of the 87 training SQLi records; (ii) the ensemble's ability to combine GBM's iterative focus on SQLi misclassifications with RF's diverse tree coverage; and (iii) the feature selection retaining Bwd Packet Length Std, which has a higher relative IG score for SQLi (whose larger database responses create characteristically variable response packet sizes) than for other attack types. XSS detection improves more modestly from Decision Tree (not reported separately in Table 4 as XSS is included in Web Attack aggregate, F1 = 0.7412) to the ensemble (F1 = 0.9012), reflecting that XSS flows are more numerous in training (652 records) and feature-discriminable than SQLi.

4.3 Feature Importance and Explainability Results

The feature-importance rankings derived from Random Forest, GBM, XGBoost, and permutation analysis are summarised in Table 5 together with interpretive descriptions of each feature's contribution to attack detection. The strong consistency observed across ranking methods reinforces the reliability of the identified discriminative features. Results in Table 5 indicate that Flow Duration is consistently ranked as the most important feature across all methods, reflecting the distinct temporal behaviours associated with different attack categories. DoS traffic is characterised by short-duration, high-frequency flows, Botnet activity exhibits long-duration periodic communication patterns, and SQL Injection attacks typically generate longer web sessions because of database query processing overhead. Destination Port, ranked between second and third across methods, captures structural attack intent by distinguishing web-service attacks from service-specific attacks such as Brute Force and Botnet activity without requiring payload-level inspection.

Table 5. Feature Importance Rankings Across Tree-Based Models and Permutation Importance (Top 10 Features)

Feature	RF Rank	GBM Rank	XGBoost Rank	Perm. Rank	Interpretive Role in Attack Detection
Flow Duration	1	1	1	1	Universal discriminator — all attacks
Destination Port	2	3	2	2	Port 80/443/8080 vs rare attack ports
Flow Bytes/s	3	2	3	3	DoS rate signature
Fwd Packet Length Mean	4	4	4	5	Payload size differs: XSS>SQLi>Normal
Fwd IAT Mean	5	5	5	4	Slow DoS vs Brute Force timing
PSH Flag Count	6	6	6	7	HTTP push pattern in Web Attacks
Init Win Bytes Forward	7	8	7	6	Handshake anomaly in port scan
Total Fwd Packets	8	7	8	8	Traffic volume indicator
ACK Flag Count	9	9	9	9	Session completion pattern
Bwd Packet Length Mean	10	10	10	10	Response size — SQLi larger than XSS

PSH Flag Count ranks 6th overall but exhibits substantially higher class-specific importance for XSS (ranked 1st; see Table 6), as HTTP POST requests carrying injected XSS payloads generate characteristic push flag patterns distinct from benign browsing sessions. The permutation importance ranking (Breiman, 2001) broadly confirms the MDI rankings but elevates Fwd IAT Mean from rank 5 to rank 4, reflecting IAT's particularly strong interaction effect with Flow Duration for distinguishing slow-rate DoS and Infiltration from benign traffic—an interaction that MDI's marginal impurity measure underestimates.

4.4 Class-Specific Feature Attribution

Class-specific feature attribution results are summarised in Table 6, where the top three discriminative features for each attack category are presented alongside their Random Forest normalised importance scores and interpretive explanations. This analysis contributes directly to IDS interpretability by providing security analysts with insight into the traffic characteristics driving model decisions. Results in Table 6 reveal feature-attribution patterns that correspond closely with the established behavioural mechanisms of each attack class. XSS attacks are strongly associated with PSH Flag Count and Fwd Packet Length Mean, reflecting the rapid transmission of short malicious HTTP POST payloads. SQL Injection traffic is primarily characterised by Fwd Packet Length Mean and Bwd Packet Length Std, indicating larger and more variable server-side responses generated by malformed database queries. DoS/DDoS attacks are dominated by Flow Bytes/s and Flow Packets/s, consistent with the high-throughput behaviour that defines resource-exhaustion attacks.

Table 6. Class-Specific Feature Attribution: Top Discriminative Features and Interpretive Explanations per Attack Class

Attack Class	Top Discriminative Features	Importance Score	Interpretive Explanation
XSS	PSH Flag Count, Fwd Packet Length Mean, Destination Port	RF Gini Impurity (normalised)=0.412	Short payloads, HTTP POST to web ports, frequent push flags
SQL Injection	Fwd Packet Length Mean, Bwd Packet Length Std, Flow Duration	RF Gini Impurity (normalised)=0.387	Larger response packets, longer sessions, variable response sizes
DoS / DDoS	Flow Bytes/s, Flow Packets/s, Fwd IAT Mean	RF Gini Impurity (normalised)=0.521	Very high packet rates, short inter-arrival times, sustained flow
Botnet	Flow Duration, Destination Port, Fwd IAT Mean	RF Gini Impurity (normalised)=0.341	Long-duration periodic polling, fixed destination ports, regular IAT
PortScan	Destination Port, SYN Flag Count, Init Win Bytes Forward	RF Gini Impurity (normalised)=0.498	Many unique ports, SYN-only connections, zero window size
Infiltration	Flow Duration, Bwd Packet Length Mean, ACK Flag Count	RF Gini Impurity (normalised)=0.289	Long slow sessions, moderate response size, completed sessions
Brute Force	Fwd IAT Mean, Destination Port, Total Fwd Packets	RF Gini Impurity (normalised)=0.356	Rapid repeated connection attempts, target SSH/FTP ports

Botnet detection relies primarily on Flow Duration and Fwd IAT Mean, capturing the characteristic periodic polling intervals of Botnet ARES's command-and-control communication—regular, spaced connections to fixed IP:port combinations that produce distinctive inter-arrival time patterns. PortScan is identified by SYN Flag Count and Init Win Bytes Forward, as SYN-only scan connections (without completing the three-way handshake) produce zero-window-size flows with elevated SYN counts—a pattern invisible to payload-based analysis but clearly visible in flow statistics. Infiltration's top features—Flow Duration, Bwd Packet Length Mean, and ACK Flag Count—reflect completed, long-duration sessions with moderate response sizes, consistent with slow-rate credential harvesting and lateral movement activities.

These class-specific feature profiles provide security analysts with actionable, feature-level explanations: an alert classified as SQL Injection can be explained to the analyst as "flagged due to unusually large and variable server response packet sizes (Bwd Packet Length Std > threshold) and longer-than-normal session duration (Flow Duration > threshold)"—an explanation directly mappable to database query behaviour and actionable for incident response.

4.5 Robustness Under Feature Noise

Robustness evaluation results under a simulated 10% random Gaussian feature-noise condition are presented in Table 7 using accuracy and macro F1-score as performance metrics. Gaussian noise with $\sigma = 0.05$ was applied to the normalised feature space to emulate realistic data degradation arising from sensor miscalibration, measurement artefacts, and label inconsistencies. Results in Table 7 indicate that Random Forest and the RF+GBM ensemble exhibit the strongest robustness, with macro F1-score reductions of only 0.0100 and 0.0142, respectively, consistent with the variance-reduction characteristics of ensemble bagging methods (Breiman, 2001). Decision Tree experiences the largest degradation among the non-naive models, with macro F1-score decreasing by 0.0678, reflecting the sensitivity of single-tree decision boundaries to noisy feature perturbations. SVM also demonstrates substantial performance decline (0.0429), attributable to the sensitivity of the RBF kernel margin to feature-scale variations. MLP exhibits moderate degradation (0.0291), suggesting partial robustness provided by dropout-based regularisation, although still weaker than ensemble tree diversity.

Table 7. Model Robustness Analysis: Performance Under 10% Random Feature Noise (Gaussian, sigma=0.05)

Model	Acc Clean (%)	Macro F1 Clean	Acc Noisy (%)	Macro F1 Noisy	F1 Drop	Robustness
Naive Bayes	82.41	0.6834	81.12	0.6621	0.0213	7th
SVM (RBF)	94.71	0.8241	91.34	0.7812	0.0429	6th
Decision Tree	97.12	0.8912	93.81	0.8234	0.0678	5th
MLP	97.63	0.9034	95.12	0.8743	0.0291	4th
Gradient Boosting	98.41	0.9287	97.12	0.9143	0.0144	2nd
Random Forest	98.74	0.9412	97.63	0.9312	0.0100	1st
XGBoost	98.93	0.9481	97.84	0.9284	0.0197	3rd
RF+GBM Voting	99.14	0.9543	98.12	0.9401	0.0142	1st=

The robustness results reinforce the model selection recommendation derived from overall accuracy: ensemble tree methods—particularly Random Forest and the RF+GBM voting ensemble—provide the best combination of high performance and noise tolerance for operational IDS deployment. This practical robustness advantage is directly relevant to real-world network environments where measurement artefacts, clock skew, and sampling variability introduce noise into CICFlowMeter-extracted features.

4.6 Comparison with Related Works

Comparative benchmarking against eight IDS studies published between 2016 and 2019 is presented in Table 8 using CICIDS2017 and comparable intrusion-detection datasets. Results in Table 8 indicate that the proposed RF+GBM ensemble achieves the highest reported classification accuracy at 99.14%, together with macro F1-score of 0.9543. The closest competing approaches are the CICIDS2017 baseline reported by Sharafaldin et al. (2018) and the XGBoost framework of Yan et al. (2018), both of which achieve lower accuracy and do not incorporate explainability analysis or class-specific web-attack evaluation. The present study further distinguishes itself by reporting separate F1-scores for XSS and SQL Injection attacks, thereby exposing minority-class performance limitations that aggregate accuracy metrics fail to reveal. Additional contributions include permutation-based feature-importance analysis for model interpretability and robustness assessment under simulated feature-noise conditions. Vinayakumar et al.'s (2019) DNN framework achieves competitive performance at 98.43% accuracy but does not provide interpretable feature-level explanations, highlighting the explanatory advantage of the proposed ensemble approach.

Table 8. Comparison of Proposed RF+GBM Voting Ensemble with Related IDS Studies (2016-2019)

Study	Method	Best Acc. (%)	Notes
Sharafaldin et al. (2018)	DT, RF, KNN, NB, ID3 (CICIDS2017 baseline)	~98.0	Binary focus; no XAI; no web-attack-specific analysis
Farnaaz & Jabbar (2016)	Random Forest (NSL-KDD)	99.67	NSL-KDD; RF only; no explainability; no web attacks
Aljawarneh et al. (2018)	J48, RF, BayesNet (WEKA)	99.40	NSL-KDD; multi-class; no XAI; no web attack focus
Hasan et al. (2016)	SVM + Relief-F FS	97.55	NSL-KDD; binary; no XAI; limited class analysis
Vinayakumar et al. (2019)	Deep DNN (CICIDS2017)	98.43	Deep learning; no XAI; binary; no feature interpretation
Yan et al. (2018)	XGBoost (CICIDS2017)	98.55	CICIDS2017; no XAI; no web-attack-specific breakdown
Peng et al. (2018)	RF + chi-square FS (CICIDS2017)	98.12	CICIDS2017; feature selection; no explainability analysis
Thaseen & Kumar (2017)	SVM + chi-square FS (NSL-KDD)	96.90	NSL-KDD; binary; limited feature analysis; no XAI
This Study (RF+GBM Voting)	RF+GBM Voting + XAI (CICIDS2017)	99.14	8-class; XAI feature/permutation importance; SQLi/XSS analysis; macro F1=0.9543

4.7 Computational Efficiency

Computational performance and deployment characteristics for all evaluated models are summarised in Table 9, including training time, inference latency, parameter count, and hardware requirements. Results in Table 9 indicate that all models except the MLP complete training without GPU acceleration, demonstrating the feasibility of deploying the proposed IDS pipeline on standard CPU-based hardware platforms typical of pre-2020 research environments. Random Forest and XGBoost achieve the most efficient ensemble-training times at 89.4 and 61.2 seconds, respectively. The RF+GBM voting ensemble requires longer training time (224.1 seconds) because both component models are trained sequentially, although this overhead remains practical for scheduled retraining cycles. All evaluated models complete inference on the 46,998-record test set in under four seconds, corresponding to throughput rates between 11,750 and 78,330 flows per second, which is sufficient for near-real-time batch intrusion analysis. Naive Bayes and Decision Tree provide the lowest inference latency at below 0.6 seconds, albeit with significantly reduced detection performance relative to the stronger ensemble models reported in Table 3.

Table 9. Computational Efficiency and Deployment Characteristics of Evaluated Models

Model	Train Time (s)	Infer Time (s)	Parameters	GPU Required	RT Feasible
Naive Bayes	1.2	0.3	N/A	~0	Yes
SVM (RBF)	198.4	3.8	N/A	~0	Marginal
Decision Tree	4.8	0.6	N/A	~0	Yes
MLP	74.3	1.2	~85,000	Optional	Yes
Gradient Boosting	134.7	2.1	N/A (trees)	~0	Yes
Random Forest	89.4	1.8	N/A (trees)	~0	Yes
XGBoost	61.2	1.4	N/A (trees)	~0	Yes
RF+GBM Voting	224.1	3.9	Combined	~0	Yes (batch)

5. Conclusion

This paper presented a comparative benchmark study of eight ML classifiers for web-attack-focused multi-class intrusion detection on the CICIDS2017 dataset, augmented by tree-based feature importance, permutation importance, and class-specific feature attribution analysis. The proposed RF+GBM soft-voting ensemble achieved 99.14% overall accuracy and a macro F1-score of 0.9543, outperforming all individual classifiers and prior published CICIDS2017 baselines. The explainability analysis revealed that Flow Duration, Destination Port, and Flow Bytes/s are universally discriminative across all attack classes, while PSH Flag Count and Fwd Packet Length Mean are the strongest specific indicators of XSS activity and response payload variability most strongly differentiates SQL Injection from benign HTTP traffic.

Three practical conclusions follow from these results. First, ensemble tree models—Random Forest and Gradient Boosting—consistently outperform single classifiers and provide the best noise robustness, making them the recommended architecture for pre-2020 production IDS deployments. Second, per-class F1 reporting—rather than aggregate accuracy—is essential for evaluating IDS utility for rare attack categories such as SQL Injection and Infiltration, where even 99% overall accuracy can mask near-zero attack detection. Third, tree-based feature importance and permutation importance provide operationally actionable explanations that security analysts can use to validate, understand, and tune ML-based IDS alerts without requiring post-hoc explanation libraries.

References

Aljawarneh, S., Aldwairi, M., & Yassein, M. B. (2018). Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. Journal of Computational Science, 25, 152-160. https://doi.org/10.1016/j.jocs.2017.03.006

Axelsson, S. (2000). Intrusion detection systems: A survey and taxonomy (Technical Report No. 99-15). Chalmers University of Technology.

Breiman, L. (2001). Random forests. Machine Learning, 45(1), 5-32. https://doi.org/10.1023/A:1010933404324

Breiman, L., Friedman, J. H., Olshen, R. A., & Stone, C. J. (1984). Classification and regression trees. Wadsworth & Brooks/Cole Advanced Books & Software.

Chandola, V., Banerjee, A., & Kumar, V. (2009). Anomaly detection: A survey. ACM Computing Surveys, 41(3), 1-58. https://doi.org/10.1145/1541880.1541882

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16, 321-357. https://doi.org/10.1613/jair.953

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785-794). ACM. https://doi.org/10.1145/2939672.2939785

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297. https://doi.org/10.1007/BF00994018

Engelen, G., Rim, V., Meert, W., & Joosen, W. (2019). Troubleshooting an intrusion detection dataset: CICIDS2017. arXiv preprint arXiv:2103.07476.

Farnaaz, N., & Jabbar, M. A. (2016). Random forest modeling for network intrusion detection system. Procedia Computer Science, 89, 213-217. https://doi.org/10.1016/j.procs.2016.06.047

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of Statistics, 29(5), 1189-1232. https://doi.org/10.1214/aos/1013203451

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of Machine Learning Research, 3, 1157-1182.

Hall, M. A. (1999). Correlation-based feature selection for machine learning (Doctoral dissertation). University of Waikato, Hamilton, New Zealand.

Hasan, M. A. M., Nasser, M., Pal, B., & Ahmad, S. (2016). Support vector machine and random forest modeling for intrusion detection system (IDS). Journal of Intelligent Learning Systems and Applications, 8(2), 48-56. https://doi.org/10.4236/jilsa.2016.82005

He, H., & Garcia, E. A. (2009). Learning from imbalanced data. IEEE Transactions on Knowledge and Data Engineering, 21(9), 1263-1284. https://doi.org/10.1109/TKDE.2008.239

Ingre, B., & Yadav, A. (2015). Performance analysis of NSL-KDD dataset using ANN. In Proceedings of the International Conference on Signal Processing and Communication Engineering Systems (SPACES) (pp. 92-96). IEEE. https://doi.org/10.1109/SPACES.2015.7058223

Korobov, M., & Lopuhin, I. (2017). eli5: A library for debugging/inspecting machine learning classifiers and explaining their predictions (Version 0.10). Retrieved from https://github.com/TeamHG-Memex/eli5

Lee, W., & Stolfo, S. J. (1998). Data mining approaches for intrusion detection. In Proceedings of the 7th USENIX Security Symposium (pp. 79-94). USENIX Association.

Lemaitre, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. Journal of Machine Learning Research, 18(17), 1-5.

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In Advances in Neural Information Processing Systems 30 (NeurIPS 2017) (pp. 4765-4774). Curran Associates.

Mitchell, T. M. (1997). Machine learning. McGraw-Hill.

OWASP Foundation. (2017). OWASP Top Ten Project 2017. Open Web Application Security Project. Retrieved from https://owasp.org/www-project-top-ten/

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 12, 2825-2830.

Peng, K., Leung, V. C. M., & Huang, Q. (2018). Clustering approach based on mini batch Kmeans for intrusion detection system over big data. IEEE Access, 6, 11897-11906. https://doi.org/10.1109/ACCESS.2018.2810267

Quinlan, J. R. (1993). C4.5: Programs for machine learning. Morgan Kaufmann.

Razzak, M. I., Imran, M., & Xu, G. (2018). Big data analytics for preventive medicine. Neural Computing and Applications, 32(9), 4417-4451. https://doi.org/10.1007/s00521-018-3476-5

Ribeiro, M. T., Singh, S., & Guestrin, C. (2016). "Why should I trust you?": Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135-1144). ACM. https://doi.org/10.1145/2939672.2939778

Salo, F., Nassif, A. B., & Essex, A. (2019). Dimensionality reduction with IG-PCA ensemble feature selection for intrusion detection system. Computer Networks, 148, 164-175. https://doi.org/10.1016/j.comnet.2018.11.010

Sharafaldin, I., Lashkari, A. H., & Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP) (pp. 108-116). SciTePress. https://doi.org/10.5220/0006639801080116

Stolfo, S. J., Fan, W., Lee, W., Prodromidis, A., & Chan, P. K. (2000). Cost-based modeling for fraud and intrusion detection: Results from the JAM project. In Proceedings of the DARPA Information Survivability Conference and Exposition (DISCEX) (Vol. 2, pp. 130-144). IEEE. https://doi.org/10.1109/DISCEX.2000.821515

Strobl, C., Boulesteix, A. L., Zeileis, A., & Hothorn, T. (2007). Bias in random forest variable importance measures: Illustrations, sources and a solution. BMC Bioinformatics, 8(1), 25. https://doi.org/10.1186/1471-2105-8-25

Tavallaee, M., Bagheri, E., Lu, W., & Ghorbani, A. A. (2009). A detailed analysis of the KDD CUP 99 data set. In Proceedings of the IEEE Symposium on Computational Intelligence for Security and Defense Applications (CISDA) (pp. 1-6). IEEE. https://doi.org/10.1109/CISDA.2009.5356528

Thaseen, I. S., & Kumar, C. A. (2017). Intrusion detection model using fusion of chi-square feature selection and multi class SVM. Journal of King Saud University - Computer and Information Sciences, 29(4), 462-472. https://doi.org/10.1016/j.jksuci.2015.12.004

Torrano-Gimenez, C., Perez-Villegas, A., & Alvarez, G. (2010). An anomaly-based web application firewall using HTTP-specific features and one-class SVM. In Proceedings of the International Conference on Internet Technology and Secured Transactions (ICITST) (pp. 1-6). IEEE.

Ustebay, S., Turgut, Z., & Aydin, M. A. (2019). Intrusion detection system with recursive feature elimination by using random forest and deep learning classifier. In Proceedings of the International Congress on Big Data, Deep Learning and Fighting Cyber Terrorism (IBIGDELFT) (pp. 71-76). IEEE. https://doi.org/10.1109/IBIGDELFT.2018.8625318

Vinayakumar, R., Alazab, M., Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep learning approach for intelligent intrusion detection system. IEEE Access, 7, 41525-41550. https://doi.org/10.1109/ACCESS.2019.2895334

Yan, J., Meng, Y., Zuo, L., & Li, T. (2018). Multiple intrusion detection algorithm using XGBoost and LightGBM. In Proceedings of the IEEE Conference on Computer Communication Workshops (INFOCOM WKSHPS) (pp. 895-900). IEEE. https://doi.org/10.1109/INFCOMW.2018.8406909