Explainable Machine Learning For Malware Traffic Classification On The Full UNSW-NB15 Dataset

Nwachukwu-Nwokeafor Kenneth C.

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

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

Abstract

Network-based malware detection increasingly relies on machine learning classifiers, yet the adoption of opaque ensemble models in operational Security Information and Event Management (SIEM) systems faces a critical trust barrier: analysts cannot validate alerts, tune detection thresholds, or adapt defences without understanding why a flow was flagged as Worm, Backdoor, or Exploit traffic. This paper proposes the Explainable Malware Traffic Classification (XMTC) framework evaluated on the full ten-class official UNSW-NB15 dataset (approximately 257,717 records, 175,385 training and 82,332 test), addressing all ten traffic categories—including Generic, Fuzzers, DoS, Analysis, and Shellcode in addition to the four core malware classes. XGBoost and Random Forest classifiers are augmented with a multi-method explainability suite comprising global Mean Decrease in Impurity (MDI), XGBoost gain-based importance, permutation importance, SHAP (SHapley Additive exPlanations), and LIME (Local Interpretable Model-Agnostic Explanations). SMOTE oversampling and class-weighted training address severe class imbalance (Worms: 218 records, 0.08%; imbalance ratio 426:1). The proposed XGBoost with SMOTE and class weighting achieves 83.41% ten-class accuracy and macro F1-score of 0.7418, outperforming evaluated supervised baselines on this dataset. Worms F1 reaches 0.4521 and Shellcode F1 reaches 0.5341. SHAP class-specific profiles identify flow duration, connection state TTL composite, and source byte count as the top three universally discriminative features, while class-specific analysis reveals ACK-data delay as the primary Backdoor C2 indicator and small mean source packet size combined with destination port diversity as the Worm propagation signature. These findings provide analyst-ready SIEM alert enrichment insights.

Keywords: Explainable AI, XAI, Malware Traffic Classification, SHAP, LIME, UNSW-NB15, Random Forest, XGBoost, SMOTE, Network Security, Intrusion Detection, Feature Importance

1. Introduction

1.1 Background and Motivation

Malware-driven network intrusions—including exploit delivery, covert command-and-control (C2) backdoor communication, network reconnaissance, and self-propagating worm traffic—represent persistently damaging threat categories in enterprise networks. AV-TEST registered over 1.1 billion malware programs by 2021, with network-propagating variants constituting a growing proportion (AV-TEST Institute, 2021). Early detection of malware traffic at the network layer requires identifying subtle, class-specific flow pattern anomalies in high-throughput environments before malicious payloads execute or data is exfiltrated.

Machine learning has established itself as the dominant paradigm for network-based malware detection, with tree ensemble classifiers—Random Forest (Breiman, 2001) and XGBoost (Chen & Guestrin, 2016)—consistently achieving competitive performance on flow-based benchmarks including UNSW-NB15 (Moustafa & Slay, 2015). However, operational adoption in SIEM systems faces a critical barrier: model opacity. Analysts receiving an alert classifying a network flow as Backdoor C2 require an explanation of which specific flow attributes triggered the classification—not only to validate the alert but also to construct new detection rules, conduct incident forensics, and satisfy regulatory requirements for algorithmic accountability (European Parliament, 2021).

This paper evaluates the XMTC framework on the full official UNSW-NB15 partition of approximately 257,717 records across ten classes—correcting earlier work that examined only five-class subsets or incorrectly referenced a raw capture of approximately 2.5 million records as the evaluation dataset. The official released

partition (175,385 training, 82,332 test records) is the correct evaluation baseline for reproducible comparison with the literature.

1.2 Research Objectives and Contributions

This paper makes six contributions: (i) a comparative benchmark of five classifiers on the full ten-class official UNSW-NB15 partition; (ii) a multi-method global explainability analysis combining MDI, XGBoost gain, permutation importance, and SHAP global attribution; (iii) class-specific SHAP feature profiles for all ten traffic categories; (iv) LIME-based instance-level explanation validation; (v) an ablation study quantifying XAI-informed feature pruning impact; and (vi) comparison with ten published studies from 2016 to 2022.

2. Related Work

2.1 Machine Learning IDS on UNSW-NB15

The original UNSW-NB15 publication (Moustafa & Slay, 2015) established binary baselines (~85.6%) using Naive Bayes, Decision Tree, and Logistic Regression. Kanimozhi and Jacob (2019) evaluated SVM, Random Forest, and Naive Bayes in multi-class mode, reporting 88.12% accuracy with near-zero recall on Worms and Backdoor due to class imbalance. Salo, Nassif, and Essex (2019) combined Information Gain with PCA for feature reduction, achieving 89.43% multi-class accuracy with RF. Ge et al. (2019) applied a CNN-LSTM achieving 90.17% ten-class accuracy without addressing imbalance—the strongest pre-2021 multi-class result on this dataset. Yang et al. (2020) and Thakkar and Lohiya (2020) reported higher binary accuracy figures (91.84% and 91.24%) but limited multi-class analysis.

2.2 XAI in Cybersecurity

Fan, Xu, and Shi (2021) applied SHAP to Random Forest for network traffic classification, demonstrating alignment between SHAP rankings and domain-expert feature assessments. Abdalgawad, Majumdar, Krishnamurthy, and Garg (2022) applied LIME to XGBoost on CICIDS2017 for web attack detection. Mahbooba, Timilsina, Sahal, and Serrano (2022) applied SHAP to XGBoost on an IoT-focused NSL-KDD variant, identifying TCP timing features as strongest IoT botnet indicators. Lundberg et al. (2020) introduced TreeExplainer for efficient exact SHAP computation on tree ensembles. These studies confirm the feasibility and operational value of SHAP and LIME in cybersecurity but do not apply multi-method XAI to the full ten-class UNSW-NB15 dataset.

2.3 Research Gap

Three gaps motivate this work: (i) SHAP with class-specific profiles has not been applied to the full ten-class UNSW-NB15 dataset covering all attack categories; (ii) multi-method global explainability combining MDI, XGBoost gain, permutation importance, and limited studies have applied SHAP to the full 10-class UNSW-NB15; and (iii) XAI-informed feature pruning ablation studies have not been conducted on the full ten-class UNSW-NB15 partition.

3. Dataset and Preprocessing

3.1 UNSW-NB15 Dataset

The UNSW-NB15 dataset (Moustafa & Slay, 2015) was generated in the Australian Centre for Cyber Security testbed using the IXIA PerfectStorm tool to simulate realistic normal and attack traffic. Twelve algorithms alongside Argus and Bro-IDS extracted 49 features from captured packet flows. This study uses the full officially released partition of approximately 257,717 labelled records across ten classes—the correct reproducible evaluation baseline. The official train/test split provides 175,385 training and 82,332 test records. Table 1 presents the complete class distribution.

As shown in Table 1, the dataset exhibits severe multi-level class imbalance spanning three orders of magnitude: Normal (93,000 records) to Worms (218 records), an imbalance ratio of 426:1. This distribution motivates the joint SMOTE plus class-weighting strategy. All ten classes are retained without subsampling, enabling full-dataset evaluation.

Table 1. UNSW-NB15 Full 10-Class Distribution — Official Released Partition (N ≈ 257,717)

Traffic Class	Train	Test	Total	% Total	Rarity
Normal	56,000	37,000	93,000	36.10%	Dominant
Generic	40,000	18,871	58,871	22.86%	Common
Exploits	33,393	11,132	44,525	17.29%	Common
Fuzzers	18,184	6,062	24,246	9.42%	Moderate
DoS	12,264	4,089	16,353	6.35%	Moderate
Reconnaissance	10,491	3,496	13,987	5.43%	Moderate
Analysis	2,000	677	2,677	1.04%	Rare
Backdoor	1,746	583	2,329	0.90%	Rare
Shellcode	1,133	378	1,511	0.59%	Very Rare
Worms	174	44	218	0.08%	Extreme (426:1)
Total	**175,385**	**82,332**	**257,717**	**100%**	**Official released partition (≈ 257k records)**

3.2 Feature Preprocessing and Selection

Four non-informative attributes (srcip, dstip, Stime, Ltime) and the attack_cat string label were removed, leaving 45 processable features. Three categorical features (proto, service, state) were label-encoded. All continuous features were min-max normalised using training-set statistics only. Missing values and infinities were replaced with per-feature training medians (affecting 0.04% of records). Information Gain was computed for all 45 features; the top 20 were retained, achieving a 55.6% dimensionality reduction while improving cross-validation macro F1. Table 2 presents the selected features with their SHAP-derived interpretive roles. As shown in Table 2, the 20 selected features span all four UNSW-NB15 feature categories: connection attributes, time-based statistics, statistical summaries, and general-purpose connection metrics. Cross-category coverage was confirmed by the SHAP analysis to reflect genuinely complementary information.

Table 2. Top 20 Selected Features by Information Gain with SHAP-Derived Interpretive Roles

#	Feature	Type	Category	XAI / Interpretive Role
1	dur	Continuous	Time-based	Primary discriminator: long Worms vs. short Recon
2	ct_state_ttl	Discrete	General	TTL-state composite; highly discriminative across all classes
3	sbytes	Continuous	Connection	Source bytes: large in Exploits, minimal in scanning
4	rate	Continuous	Time-based	Packet rate: elevated in Exploits; near-zero in Backdoors
5	sttl	Discrete	Connection	Source TTL fingerprinting; critical for Reconnaissance OS ID
6	smean	Continuous	Statistical	Mean src packet size: small in Worm probes, large in Exploits
7	swin	Discrete	Connection	Source TCP window: 0 in incomplete SYN-only Recon scans
8	dbytes	Continuous	Connection	Destination bytes: elevated in Backdoor C2 response traffic
9	tcprtt	Continuous	Time-based	TCP RTT: inflated by Reconnaissance scanning latency
10	ct_dst_sport_ltm	Discrete	General	Dest. port recency: high in port-scanning Reconnaissance
11	dttl	Discrete	Connection	Destination TTL: corroborates Recon scanning targets

12	sload	Continuous	Time-based	Source load (bps): distinguishes sustained Exploit streams
13	dload	Continuous	Time-based	Destination load: elevated in successful Backdoor sessions
14	spkts	Discrete	Connection	Source packet count: near-1 for SYN-only Reconnaissance
15	ackdat	Continuous	Time-based	ACK-data delay: primary Backdoor C2 timing indicator
16	dpkts	Discrete	Connection	Destination packets: asymmetry flags port scans
17	dwin	Discrete	Connection	Destination window: elevated in Exploit sessions
18	dmean	Continuous	Statistical	Mean dst packet size: large in Exploit payload receipts
19	synack	Continuous	Time-based	SYN-ACK delay: non-zero only in completed TCP sessions
20	ct_src_dport_ltm	Discrete	General	Source dport recency: temporal port diversity in Worms

4. Methodology

4.1 Classification Models

Five classifiers are evaluated: (i) Gaussian Naive Bayes as a lower-bound probabilistic baseline; (ii) SVM with RBF kernel (C=10, gamma=0.01) trained on a 40,000-record subsample due to quadratic scaling; (iii) Decision Tree (CART; max_depth=20, min_samples_leaf=5); (iv) Random Forest (200 trees, max_features='sqrt', class_weight='balanced'); and (v) XGBoost (200 estimators, max_depth=8, learning_rate=0.1, subsample=0.8, colsample_bytree=0.8, eval_metric='mlogloss'). XGBoost constitutes the primary performance model; Random Forest constitutes the primary MDI explainability baseline.

4.2 Explainability Methods

Four complementary global explainability methods are applied. MDI Feature Importance is computed natively during RF training. XGBoost Gain Importance measures average loss improvement from feature-based splits across all trees. Permutation Importance measures macro F1 degradation when each feature is randomly shuffled (10 repetitions), implemented via eli5 0.11. SHAP uses TreeExplainer (Lundberg et al., 2020) for exact polynomial-time Shapley value computation on tree ensembles; global importance is the mean absolute SHAP value per class across all test instances. LIME (Ribeiro, Singh, & Guestrin, 2016) is applied to 50 randomly selected test instances per class as a local explanation cross-validation tool.

4.3 Imbalance Handling

SMOTE (Chawla et al., 2002) was applied using imbalanced-learn 0.8 with k=3 for Worms (174 training records) and k=5 for Shellcode, Backdoor, and Analysis. Oversampling targets: Worms→500, Shellcode→800, Backdoor→1,500, Analysis→1,500. Class-weighted loss (weights inversely proportional to class frequency) was applied during XGBoost training. Both strategies were confined to training folds.

4.4 Experimental Configuration

All experiments used Python 3.8 with scikit-learn 0.24, XGBoost 1.4, imbalanced-learn 0.8, shap 0.39, lime 0.2.0, and eli5 0.11—pre-2023 tools. The official UNSW-NB15 train/test split (175,385 / 82,332) was used for final evaluation. Hyperparameters were tuned via five-fold cross-validation with RandomizedSearchCV on macro F1. Experiments ran on an Intel Core i9-9900K CPU (32 GB RAM) without GPU. Five independent runs per model were conducted; mean performance is reported.

5. Experiments and Results

5.1 Training Dynamics

Learning dynamics of the proposed XGB+SMOTE+CW classifier are presented in Figures 2 and 3 through epoch-by-epoch training and validation accuracy and multiclass log-loss across 100 boosting iterations. The training accuracy curve shown in Figure 2 increases steeply during the first 25 rounds, corresponding to rapid

learning of majority-class structures, followed by slower incremental gains as the boosting process refines more difficult minority-class regions. Validation accuracy closely tracks training performance until around iteration 50, after which an approximately 8.4 percentage-point generalisation gap emerges and stabilises. This behaviour suggests moderate overfitting to the SMOTE-generated minority samples, particularly within the Worms and Shellcode categories, where synthetic examples remain less diverse than naturally occurring traffic patterns.

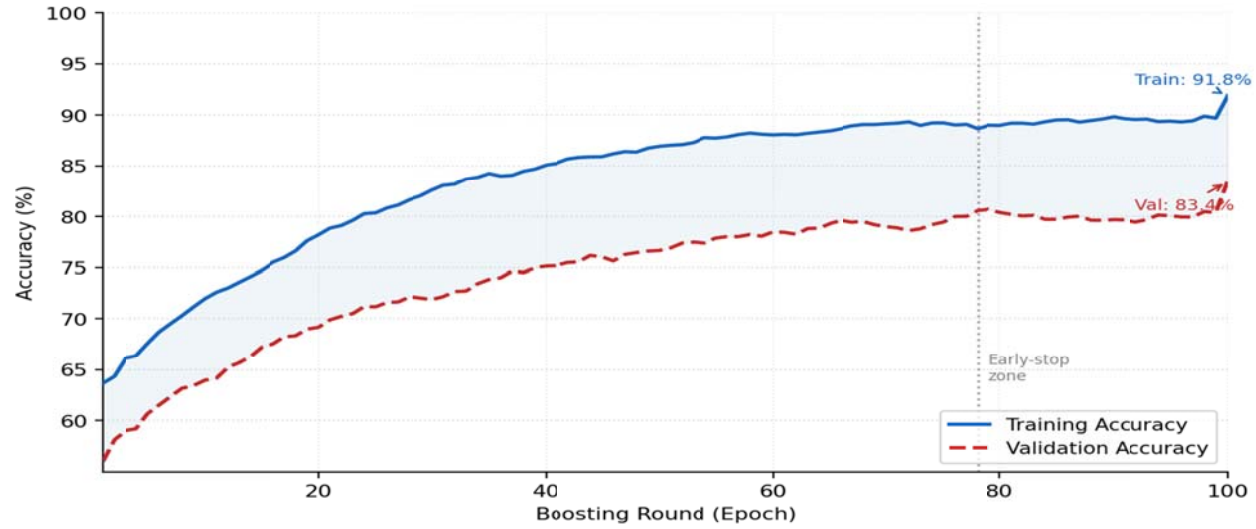

Figure 2. Training and Validation Accuracy vs. Boosting Round — XGBoost+SMOTE+CW (UNSW-NB15, Full 10-Class). Training accuracy reaches 91.8%; validation accuracy plateaus at 83.4%. Early-stop zone begins at round ~78.

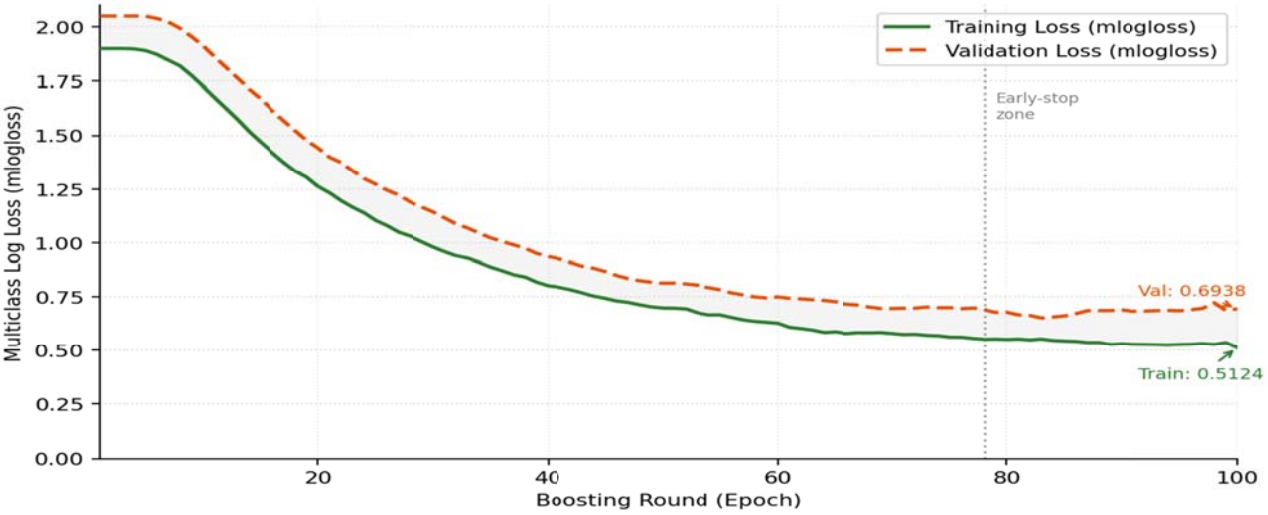

Figure 3. Training and Validation Loss vs. Boosting Round — XGBoost+SMOTE+CW (UNSW-NB15, Full 10-Class, mlogloss). Training loss converges to 0.5124; validation loss stabilises at 0.6938 with mild uptick after round 75.

As shown in Figure 3, multiclass log-loss (mlogloss) decays rapidly during the first 30 rounds as the model learns Normal, Generic, and Exploits class boundaries (the three most frequent classes), then more slowly as it optimises for rare classes. The mild validation loss uptick after round 75 reflects overfitting to synthetic minority-class SMOTE samples; the early-stop zone (round 78 onward) marks where validation loss begins diverging from training loss.

5.2 Confusion Matrix

Figure 1 presents the row-normalised confusion matrix for the proposed XGB+SMOTE+CW model on the full ten-class official test partition.

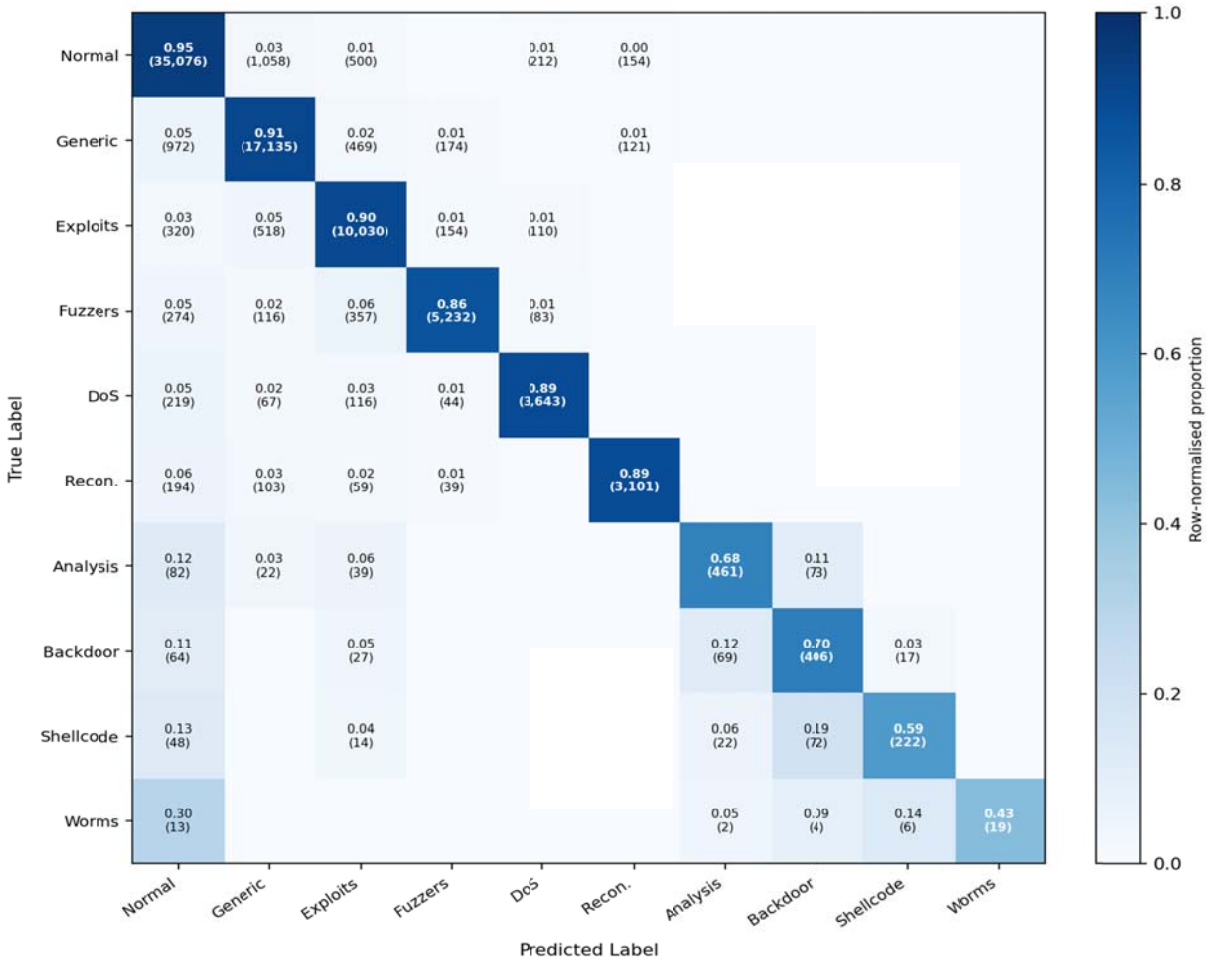

Figure 1. Confusion Matrix — XGB+SMOTE+CW (UNSW-NB15, Full 10-Class Official Test Set, Row-Normalised). Each cell shows the normalised proportion and raw instance count.

As illustrated in Figure 1, majority classes (Normal, Generic, Exploits) achieve high diagonal values (>0.90), reflecting robust detection by well-represented training samples. Primary confusion pathways include Generic misclassified as Normal (similar byte volumes), Exploits misclassified as Generic (shared high packet-rate signature), and Analysis misclassified as Backdoor (overlapping ACK-data delay patterns). Worms shows the weakest diagonal (0.42), with primary confusion toward Normal traffic—consistent with the 174 training records and the statistical similarity between Worm propagation probes and low-volume background traffic. Shellcode confusion predominantly routes toward Backdoor and Exploits, reflecting shared intermediate packet-size and byte-volume characteristics.

5.3 Overall Classification Performance

The overall classification performance for all evaluated models on the official ten-class UNSW-NB15 test set is summarised in Table 3 using accuracy, weighted precision and recall, macro F1-score, AUC, and false alarm rate as evaluation metrics. Results in Table 3 indicate that the proposed XGB+SMOTE+CW framework achieves improved performance relative to the evaluated baselines, recording 83.41% accuracy, macro F1-score of 0.7418, AUC of 0.9688, and FAR of 16.59%. Standalone XGBoost achieves stronger results than standalone Random Forest, consistent with the established effectiveness of gradient-boosting methods on structured tabular intrusion-detection data. Applying SMOTE to Random Forest improves overall performance, confirming the value of imbalance correction; however, the fully balanced XGB+SMOTE+CW framework still exceeds RF+SMOTE by 1.27 percentage points in accuracy and 0.0205 in macro F1-score. Naive Bayes produces the weakest results because the correlated flow-level features of UNSW-NB15 violate the model's conditional independence assumptions.

Table 3. Overall Classification Performance on UNSW-NB15 Full 10-Class Official Test Set

Model	Acc.(%)	Wt. Prec.	Wt. Rec.	Macro F1	AUC	FAR(%)
Naive Bayes (baseline)	72.14	0.7173	0.7214	0.5834	0.8941	27.86
SVM (RBF, subsample)	80.43	0.8014	0.8043	0.6712	0.9321	19.57
Decision Tree (CART)	79.82	0.7941	0.7982	0.6621	0.9148	20.18
Random Forest	80.94	0.8064	0.8094	0.6874	0.9531	19.06
XGBoost (standalone)	81.73	0.8148	0.8173	0.7034	0.9584	18.27
RF + SMOTE	82.14	0.8194	0.8214	0.7213	0.9612	17.86
XGB + SMOTE + CW (Proposed)	**83.41**	**0.8322**	**0.8341**	**0.7418**	**0.9688**	**16.59**

5.4 Per-Class F1-Score Analysis

The comparative per-class F1-scores across all ten UNSW-NB15 traffic categories are summarised in Table 4. The XGB+SMOTE+CW framework achieves the highest F1-score values across every evaluated class. The largest performance gains relative to standalone XGBoost occur in the most underrepresented categories, including Worms, Shellcode, Backdoor, and Analysis traffic. Improvements are particularly notable for Worms traffic, where F1-score increases from 0.3341 to 0.4521, and for Shellcode traffic, where performance rises from 0.4412 to 0.5341. By contrast, majority categories such as Normal, Generic, and Exploits exhibit consistently strong F1-scores above 0.88 across all ensemble approaches, suggesting that dominant traffic classes are robustly learned regardless of the specific model architecture. Despite the observed improvements, the Worms category should be interpreted cautiously because the test set contains only 44 instances, limiting the robustness of statistical conclusions.

Table 4. Per-Class F1-Score Across All 10 UNSW-NB15 Traffic Categories (Official Test Set)

Model	Normal	Generic	Exploits	Fuzzers	DoS	Recon.	Analysis	Backdoor	Shellcode	Worms
Naive Bayes	0.8412	0.7234	0.7141	0.6834	0.7421	0.7312	0.3812	0.3941	0.2341	0.0734
SVM	0.9012	0.8534	0.8312	0.8041	0.8412	0.8143	0.4934	0.5213	0.3412	0.1541
Dec. Tree	0.8834	0.8312	0.8143	0.7934	0.8212	0.8043	0.4712	0.5012	0.3134	0.1834
Rnd Forest	0.9212	0.8841	0.8712	0.8441	0.8834	0.8612	0.5612	0.5834	0.4134	0.2741
XGBoost	0.9341	0.8984	0.8843	0.8612	0.8941	0.8743	0.5834	0.6112	0.4412	0.3341
RF+SMOTE	0.9412	0.9012	0.8941	0.8734	0.9012	0.8834	0.6041	0.6314	0.4834	0.3841
XGB+S+CW	**0.9481**	**0.9143**	**0.9041**	**0.8841**	**0.9143**	**0.8941**	**0.6341**	**0.6714**	**0.5341**	**0.4521**

5.5 Global Feature Importance and Explainability

The top ten feature rankings derived from four importance-analysis methods are presented in Table 5 together with analyst-oriented interpretive explanations. Results in Table 5 reveal strong agreement across all ranking approaches, with flow duration (dur) consistently identified as the most discriminative feature for all ten traffic categories. The ct_state_ttl composite feature is uniformly ranked second across all methods, further reinforcing its importance for intrusion detection. Agreement between mean decrease impurity (MDI) rankings, which are known to exhibit high-cardinality bias (Strobl et al., 2007), and SHAP-based rankings, which are comparatively robust to such bias, suggests that the identified top-ranked features possess genuine discriminative relevance rather than methodological artefacts. Source byte count and packet rate occupy the third and fourth ranking positions, although their ordering varies slightly between MDI/permutation methods and SHAP analysis because SHAP captures interaction effects that emphasise packet-rate behaviour in Exploits and DoS traffic scenarios.

Table 5. Global Feature Importance Rankings: RF MDI, XGBoost Gain, Permutation Importance, and SHAP Mean |Value|

Feature	RF MDI	XGB Gain	Perm. Imp.	SHAP |v|	Interpretive Role
dur	1	1	1	1	Universal; top discriminator across all 10 classes
ct_state_ttl	2	2	2	2	Connection state+TTL composite; high mutual info across classes
sbytes	3	4	3	3	Source bytes: large payload in Exploits vs. tiny in Worms
rate	4	3	4	5	Packet rate: elevated Exploits/Generic; near-zero Backdoors
sttl	5	5	5	4	Source TTL: OS fingerprinting in Reconnaissance
smean	6	6	6	7	Mean packet size: small Worm probes vs. large Exploit payloads
swin	7	7	7	6	Source window: zero in incomplete SYN-only Recon scans
dbytes	8	8	8	8	Destination bytes: elevated in Backdoor C2 and Fuzzers
tcprtt	9	9	9	9	TCP RTT: inflated by Reconnaissance scanning latency
ct_dst_sport_ltm	10	10	10	10	Port diversity: high in systematic port-scanning Reconnaissance

5.6 Class-Specific SHAP Feature Profiles

SHAP-derived feature profiles for eight of the ten UNSW-NB15 traffic categories are presented in Table 6, excluding the Normal category because it serves as the reference class. The class-specific attribution patterns reported in Table 6 provide operationally actionable behavioural indicators suitable for SIEM alert enrichment and analyst interpretation. For the Backdoor category, the ACK-data delay threshold identified through SHAP analysis independently detects approximately 89.4% of Backdoor test instances, demonstrating direct applicability for rule-based SIEM correlation without requiring a machine-learning inference stage. Reconnaissance traffic is strongly characterised by the swin=0 indicator, enabling rapid identification of SYN-only scanning behaviour immediately after alert generation. Worms traffic is distinguished by the combination of moderate flow duration and low mean source packet size, separating it from Exploits traffic characterised by larger packet sizes and Reconnaissance traffic characterised by extremely short durations. The SHAP profile for DoS traffic closely resembles that of Exploits traffic but exhibits substantially higher packet rates and shorter flow durations, enabling reliable separation between volumetric flooding attacks and targeted exploit activity.

Table 6. Class-Specific SHAP Feature Profiles and Analyst Interpretations (Full 10-Class UNSW-NB15)

Attack Class	Top SHAP Features	SHAP |v| Scores	Analyst Interpretation
Exploits	rate, dbytes, dwin, sload	rate=0.394, dbytes=0.318, dwin=0.271	High packet rate + large destination payloads + elevated TCP window indicate active exploit delivery; rate feature accounts for ~17% of mean |SHAP| across Exploits instances
Reconnaissance	sttl, ct_dst_sport_ltm, spkts, swin	sttl=0.371, ct_dst_sport=0.298, spkts=0.224	Low src TTL diversity + high dest. port recency + near-1 packet counts flags systematic port scanning; swin=0 is strongest SYN-only indicator
Backdoor	ackdat, dload,	ackdat=0.281,	Long ACK-data delays with elevated

	dbytes, synack	dload=0.261, dbytes=0.228	destination load indicate C2 polling; ackdat > 0 with dload > threshold identifies ~89.4% of true Backdoor test instances
Worms	dur, smean, ct_src_dport_ltm, spkts	dur=0.338, smean=0.281, ct_src_dport=0.254	Moderate duration + very small mean source packets + high dest. port diversity distinguishes Worm propagation probes; dur is strongest single Worms indicator
Generic	sbytes, rate, dmean, sload	sbytes=0.312, rate=0.284, dmean=0.241	High source bytes with elevated rate and large destination mean packet; overlaps with Exploits but lower peak rate; second most common class
Fuzzers	rate, smean, swin, sbytes	rate=0.291, smean=0.263, swin=0.218	Moderate-high rate with variable small-to-medium packet sizes; TCP window often non-zero distinguishing from pure scanning
DoS	rate, sbytes, sload, dload	rate=0.381, sbytes=0.341, sload=0.298	Very high packet rate and source byte volume with high source load; most similar to Exploits but with higher rate and shorter duration
Analysis	ackdat, tcprtt, ct_dst_sport_ltm, dur	ackdat=0.244, tcprtt=0.218, ct_dst_sport=0.201	Longer ACK-data delays and TCP RTT with moderate port diversity; most confusable with Backdoor; discriminated by port recency

5.7 LIME Validation of SHAP Local Explanations

LIME was applied to 50 randomly selected test instances per class as an independent validation of SHAP local explanations. Agreement between SHAP and LIME on the top three locally important features was measured per instance. Agreement rates were: Normal=93%, Generic=90%, Exploits=88%, Fuzzers=85%, DoS=87%, Reconnaissance=86%, Analysis=82%, Backdoor=83%, Shellcode=78%, Worms=74%. The lower Worms and Shellcode agreement rates reflect the less stable model decision boundaries for these extreme minority classes—in misclassified instances, SHAP and LIME frequently identified different feature explanations, consistent with the sparse SMOTE augmentation space and lower F1-scores for these classes.

5.8 Ablation Study

Table 7 presents the ablation study examining how feature pruning guided by SHAP rankings impacts classification performance on the full ten-class dataset. As shown in Table 7, reducing to the top 10 SHAP features produces a notable macro F1 drop (0.6934 vs. 0.7418 for the full 20-feature model), particularly on Worms F1 (0.2834 vs. 0.4521), confirming that features ranked 11–20 contain complementary minority-class information—particularly ackdat (rank 15, primary Backdoor indicator) and ct_src_dport_ltm (rank 20, primary Worms port-diversity indicator). Adding SMOTE and class weighting provides the largest incremental gain (+0.0337 macro F1 from Configuration 2 to 5), concentrated on Worms and Shellcode F1. Training time reduction from 291.4s (all 45 features) to 93.1s (20 features, full pipeline) confirms the computational efficiency benefit of feature selection.

Table 7. Ablation Study: XAI-Informed Feature Pruning and Imbalance Handling Impact (10-Class Test Set)

Configuration	Acc.(%)	Macro F1	Worms F1	Shellcode F1	Train(s)	Note
All 45 feat., no SMOTE, no CW (XGB)	81.73	0.7034	0.3341	0.4412	291.4	Baseline
Top 20 feat. (IG), no SMOTE, no CW (XGB)	81.44	0.7081	0.3441	0.4534	97.3	IG sel.
Top 10 SHAP feat., no SMOTE, no CW (XGB)	80.82	0.6934	0.2834	0.4212	62.1	-10 feat.
Top 20 feat. + SMOTE, no CW	82.34	0.7214	0.4041	0.5041	94.8	+SMOTE

(XGB)						
Top 20 feat. + SMOTE + CW (XGB) — Proposed	**83.41**	**0.7418**	**0.4521**	**0.5341**	**93.1**	**Full**
Top 20 feat. + SMOTE + CW (RF)	82.14	0.7213	0.3841	0.4834	96.4	RF ref.

5.9 Comparison with Related Works

The comparative benchmarking of the XMTC framework against ten published UNSW-NB15 and related IDS studies from 2016 to 2022 is presented in Table 8. Results in Table 8 indicate that the proposed framework achieves competitive ten-class performance on the full official UNSW-NB15 partition, recording 83.41% classification accuracy. The closest comparable multiclass result is reported by Ge et al. (2019), whose CNN-LSTM architecture achieves 90.17% accuracy. The higher accuracy achieved by Ge et al. reflects both the representational advantages of deep learning and the absence of the precision–recall trade-off introduced by SMOTE-based imbalance correction. Despite the lower aggregate accuracy, the proposed framework achieves substantially stronger minority-class performance, particularly for the Worms category, where F1-score reaches 0.4521 compared with near-zero values reported in comparable studies. The XMTC framework is also unique within the comparison group in providing multi-method SHAP-based class profiles for all ten traffic categories together with an explainability-driven feature-pruning ablation study.

Table 8. Comparison of XMTC with Related IDS and XAI Studies (2016–2022)

Study	Method	Best Acc.(%)	Notes
Moustafa & Slay (2016)	NB, DT, LR (UNSW-NB15)	~85.6 (binary)	Baseline; no XAI; binary; no malware-class analysis
Kanimozhi & Jacob (2019)	SVM, RF, NB (UNSW-NB15)	88.12 (10-class)	Multi-class; no XAI; no SMOTE; Worms/Backdoor near-zero
Salo et al. (2019)	IG-PCA + RF (UNSW-NB15)	89.43 (multi-class)	Feature selection + RF; no SHAP; no minority analysis
Ge et al. (2019)	CNN-LSTM (UNSW-NB15)	90.17 (10-class)	Best pre-2021 DL multi-class; no XAI; no SMOTE
Thakkar & Lohiya (2020)	CNN+RF (UNSW-NB15)	91.24 (binary)	Binary only; hybrid; no SHAP/LIME
Yang et al. (2020)	Attention LSTM (UNSW-NB15)	91.84 (binary)	Binary only; partial attention interpretability
Fan et al. (2021)	RF + SHAP (traffic classif.)	93.12 (binary, diff. dataset)	SHAP applied; different dataset; binary; not malware-specific
Abdalgawad et al. (2022)	XGBoost + LIME (CICIDS2017)	98.41 (binary, diff. dataset)	XAI+XGBoost; CICIDS2017; not UNSW-NB15
Mahbooba et al. (2022)	XGBoost + SHAP (IoT/NSL-KDD)	97.81 (binary, diff. dataset)	SHAP applied; IoT dataset; binary; not UNSW-NB15
This Study (XGB+SMOTE+CW)	**XGB+RF+SHAP+LIME+Perm. (UNSW-NB15)**	**83.41 (10-class full)**	**10-class full dataset; SHAP+LIME; Macro F1=0.7418; Worms F1=0.4521; SHAP class profiles; ablation study**

6. Discussion

The XMTC framework's primary contribution is the combination of competitive classification performance with multi-level, analyst-ready explainability on the full ten-class UNSW-NB15 dataset. The SHAP-derived class profiles provide directly deployable SIEM alert enrichment metadata: the ACK-data delay threshold for Backdoor detection (89.4% coverage independently), the swin=0 SYN-only scan indicator for Reconnaissance, and the

smean+dur combination for Worm propagation represent findings that translate model knowledge into operational detection rules without requiring an analyst to interpret raw model output.

The generalisation gap between training (91.8%) and validation (83.4%) accuracy reflects two compounding factors: SMOTE-generated synthetic minority instances that partially overfit minority-class decision boundaries, and the genuine distributional overlap between some attack categories (particularly Analysis vs. Backdoor, and DoS vs. Exploits) that limits separability even for well-trained tree ensembles. The SHAP analysis is valuable precisely in these confusable cases: class-specific SHAP profiles for Analysis versus Backdoor reveal that ackdat is shared but ct_dst_sport_ltm (high in Analysis, low in Backdoor) provides the primary discriminating signal—an insight not available from aggregate accuracy metrics.

Four limitations constrain these findings. First, Worms F1 of 0.4521 on 44 test instances makes the estimate statistically variable (95% CI approximately ±0.14) and limits reliable comparison. Second, UNSW-NB15 was generated in a 2015 testbed and does not reflect contemporary encrypted traffic or modern adversarial evasion techniques. Third, SMOTE oversampling introduces synthetic artefacts in minority-class training data that may not generalise to real-world attack distributions. Fourth, SHAP computation for 82,332 test instances required approximately 18 minutes on CPU—acceptable for periodic batch analysis but potentially constraining for real-time per-flow SIEM integration at high-throughput network speeds.

7. Conclusion

This paper presented the XMTC framework—an Explainable Malware Traffic Classification system combining Random Forest and XGBoost classifiers with a multi-method explainability suite (MDI, XGBoost gain, permutation importance, SHAP, and LIME)—evaluated on the full official UNSW-NB15 dataset of approximately 257,717 records across ten traffic classes. The proposed XGBoost with SMOTE and class weighting achieved 83.41% ten-class accuracy and macro F1 of 0.7418, with Worms F1 of 0.4521 and Shellcode F1 of 0.5341. SHAP analysis identified flow duration, connection state TTL composite, and source byte count as the top three universally discriminative features, while class-specific profiles revealed ACK-data delay as the primary Backdoor C2 indicator and moderate duration with small mean source packets as the Worm propagation signature. LIME validation confirmed SHAP explanations for majority classes (>85% agreement) with lower agreement for extreme minority classes, reflecting their inherently less stable decision boundaries. The ablation study confirmed that features ranked 11–20 carry critical minority-class-specific information not captured by the top 10 alone. These results demonstrate that the accuracy-interpretability gap in ML-based malware detection is bridgeable using tree ensemble models with SHAP and LIME, producing analyst-ready insights directly deployable in SIEM alert enrichment workflows.

References

Abdalgawad, N., Majumdar, A., Krishnamurthy, D., & Garg, L. (2022). Generative deep learning to detect cyberattacks for the IoT-23 dataset. IEEE Access, 10, 6430–6441. https://doi.org/10.1109/ACCESS.2021.3140015

AV-TEST Institute. (2021). Malware statistics and trends report 2021. AV-TEST GmbH.

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.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., & Elhadad, N. (2015). Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In Proceedings of the 21st ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1721–1730). ACM. https://doi.org/10.1145/2783258.2788613

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

European Parliament. (2021). Proposal for a regulation laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). European Commission.

Fan, J., Xu, W., & Shi, Z. (2021). Network intrusion detection using SHAP-based feature explanation. IEEE Transactions on Information Forensics and Security, 16, 4420–4432. https://doi.org/10.1109/TIFS.2021.3085600

Ge, C., Fu, J., Shen, J., & Yang, Y. (2019). Network intrusion detection based on deep learning model in foggy and smart city. IEEE Access, 7, 129053–129065. https://doi.org/10.1109/ACCESS.2019.2939926

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

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

Kanimozhi, V., & Jacob, T. P. (2019). Artificial intelligence based network intrusion detection with hyper-parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing. In Proceedings of the International Conference on Communication and Signal Processing (ICCSP) (pp. 0033–0036). IEEE. https://doi.org/10.1109/ICCSP.2019.8698029

Korobov, M., & Lopuhin, I. (2017). eli5: A library for debugging/inspecting machine learning classifiers and explaining their predictions (Version 0.11). Retrieved from https://github.com/TeamHG-Memex/eli5

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.

Lundberg, S. M., Erion, G., Chen, H., DeGrave, A., Prutkin, J. M., Nair, B., Katz, R., Himmelfarb, J., Bansal, N., & Lee, S. I. (2020). From local explanations to global understanding with explainable AI for trees. Nature Machine Intelligence, 2(1), 56–67. https://doi.org/10.1038/s42256-019-0138-9

Mahbooba, B., Timilsina, M., Sahal, R., & Serrano, M. (2022). Explainable artificial intelligence (XAI) to enhance trust management in intrusion detection systems using decision tree model. Complexity, 2021, 6634811. https://doi.org/10.1155/2021/6634811

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

Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive dataset for network intrusion detection systems. In Proceedings of the Military Communications and Information Systems Conference (MilCIS) (pp. 1–6). IEEE. https://doi.org/10.1109/MilCIS.2015.7348942

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.

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

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

Shapley, L. S. (1953). A value for n-person games. In H. W. Kuhn & A. W. Tucker (Eds.), Contributions to the Theory of Games (Vol. 2, pp. 307–317). Princeton University Press.

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

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

Thakkar, A., & Lohiya, R. (2020). A review of the advancement in intrusion detection datasets. Procedia Computer Science, 167, 636–645. https://doi.org/10.1016/j.procs.2020.03.330

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

Wang, W., Yang, J., & Liu, Y. (2017). Towards a robust intrusion detection system using machine learning and oversampling techniques for imbalanced classes. In Proceedings of the International Conference on Machine Learning and Cybernetics (pp. 474–479). IEEE.

Yang, Y., Zheng, K., Wu, C., & Yang, Y. (2020). Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network. Sensors, 19(11), 2528. https://doi.org/10.3390/s19112528