Science and Technology Publishing (SCI & TECH)
ISSN: 2632-1017
Vol. 4 Issue 1, January - 2020

Feature Selection And Ensemble Learning For Multi-Class Intrusion Detection In Imbalanced Network Traffic

Nwachukwu-Nwokeafor Kenneth C.

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

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

Abstract

The rapid escalation of network-based attacks and the growing complexity of modern cyber threats present substantial challenges for contemporary intrusion detection systems (IDS). Machine learning-based IDS have emerged as the dominant paradigm for anomaly-based detection, yet two persistent challenges constrain their operational effectiveness: the high dimensionality of flow-based network features and the severe class imbalance between prevalent benign and common attack traffic and rare, high-impact attack categories such as Botnet, Web Attacks, and Infiltration. The CICIDS2017 dataset, generated under realistic network conditions with 78 flow-based features and eight traffic classes, serves as the evaluation benchmark in this study. This paper proposes an ensemble-based multi-class IDS framework that integrates a hybrid Information Gain plus Correlation-based Feature Selection (IG+CFS) pipeline with a stacking ensemble of Random Forest and XGBoost classifiers, augmented by SMOTE oversampling to address class imbalance. The hybrid feature selection reduces the 78-feature CICIDS2017 space to 15 highly discriminative attributes, an 80.8% reduction, while improving macro F1-score and reducing training time. The proposed RF+XGBoost stacking ensemble with SMOTE achieves 99.34% overall accuracy and a macro F1-score of 0.9512, outperforming all individual baselines and prior published CICIDS2017 studies. Per-class analysis demonstrates particular improvements on minority attack categories, with Infiltration F1 increasing from 0.2143 (Decision Tree baseline) to 0.6743.

Keywords: *Intrusion Detection System, CICIDS2017, Feature Selection, Information Gain, Ensemble Learning, XGBoost, Random Forest, Stacking, SMOTE, Class Imbalance, Multi-Class Classification*

1. Introduction

1.1 Background

The continuous expansion of networked systems and the proliferation of internet-connected devices have fundamentally altered the threat landscape for organisations of all sizes. Network-based attacks—encompassing Distributed Denial-of-Service (DDoS) campaigns, port scanning, brute-force credential attacks, web application exploits, botnet command-and-control traffic, and network infiltration—impose substantial operational, financial, and reputational costs. McAfee (2018) estimated that global cybercrime costs exceeded USD 600 billion in 2017, with network intrusions contributing a significant portion. Intrusion detection systems (IDS) serve as a critical defensive layer by monitoring network traffic and identifying malicious patterns before they can cause irreversible harm.

Traditional signature-based IDS, while effective against known attack patterns, cannot detect novel or zero-day threats and incur substantial maintenance overhead as attack taxonomies evolve (Axelsson, 2000). Machine learning-based anomaly detection offers greater flexibility, and the availability of realistic network traffic datasets, particularly the CICIDS2017 dataset generated by the Canadian Institute for Cybersecurity, has catalysed a growing body of ML-based IDS research (Sharafaldin, Lashkari, & Ghorbani, 2018).

1.2 Problem Statement

Two interconnected challenges constrain the effectiveness of ML-based IDS on modern network traffic datasets. First, flow-based feature extraction tools such as CICFlowMeter generate 78 to 80 per-flow statistical features, many of which are redundant, correlated, or carry minimal discriminative information for specific attack classes (Lashkari, Draper-Gil, Mamun, & Ghorbani, 2017). Training high-complexity ensemble models on the full 78-feature space incurs substantial computational overhead and risks overfitting to noise features, degrading generalisation to unseen attack patterns. Second, real-world network traffic is inherently imbalanced: benign flows dominate (over 80% of records in CICIDS2017), while critical rare attack categories such as Botnet (0.07%), Web Attacks (0.08%), and Infiltration (<0.01%) are drastically underrepresented. Standard classifiers optimising overall accuracy will consistently fail on these minority classes, precisely the categories that represent the most sophisticated and damaging real-world threats (He & Garcia, 2009).

1.3 Research Motivation

Research conducted between 2018 and 2019 consistently demonstrates that feature selection reduces IDS training time and improves generalisation by removing noise and redundant attributes (Sharafaldin et al., 2018; Peng, Ding, Jiang, & Wang, 2018). Simultaneously, ensemble methods, particularly gradient-boosted tree ensembles, have achieved state-of-the-art results on tabular classification tasks, with XGBoost (Chen & Guestrin, 2016) establishing itself as the dominant algorithm for structured network traffic classification by 2019. The combination of rigorous feature selection, ensemble learning, and principled imbalance handling within a unified multi-class framework has been insufficiently explored on CICIDS2017 prior to 2020, motivating the present study.

1.4 Aims, Objectives, and Contributions

This study develops a scalable and balanced multi-class intrusion detection framework for the CICIDS2017 dataset by integrating feature selection, ensemble learning, and class imbalance handling. It evaluates multiple feature selection techniques and classifiers, assesses the impact of SMOTE on minority-class detection, and reports both aggregate and per-class performance metrics. The key contributions include a hybrid IG+CFS feature selection method that reduces dimensionality from 78 to 15 features, a stacking ensemble combining Random Forest and XGBoost with a Logistic Regression meta-learner, and a comprehensive evaluation framework that highlights improved detection of rare attack classes in a seven-class setting.

2. Related Work

2.1 Machine Learning Approaches for IDS

Machine learning has been applied to intrusion detection for over two decades, with early work focused on the KDD Cup 1999 and NSL-KDD benchmarks (Lee & Stolfo, 1998; Tavallaee, Bagheri, Lu, & Ghorbani, 2009). Decision trees offer interpretable, fast-training detection models but are prone to overfitting on noisy flow features (Quinlan, 1993). Support Vector Machines achieve strong binary detection performance but scale poorly to large datasets and multi-class settings (Cortes & Vapnik, 1995). k-Nearest Neighbours is effective in low-dimensional spaces but incurs O(n) inference cost that renders it impractical for high-throughput IDS deployment (Cover & Hart, 1967). With the release of CICIDS2017, these classical methods have been re-evaluated in a more realistic, high-dimensional setting, with most studies reporting binary or partial multi-class results (Sharafaldin et al., 2018; Lashkari et al., 2017).

2.2 Feature Selection in IDS

Feature selection reduces input dimensionality, decreases training time, and can improve model generalisation by removing noisy or redundant attributes. Filter methods, information gain (Quinlan, 1993), chi-square statistics, and Pearson correlation, evaluate features independently of a downstream classifier and are computationally efficient (Guyon & Elisseeff, 2003). Correlation-based Feature Selection (CFS) extends simple filtering by explicitly penalising features that are highly correlated

with already-selected attributes, promoting a diverse and non-redundant subset (Hall, 1999). Principal Component Analysis (PCA) applies a linear transformation to maximise explained variance across the reduced component space (Jolliffe, 2002), though at the cost of interpretability. Peng et al. (2018) demonstrated that chi-square feature selection on CICIDS2017 reduced features from 78 to 20 while maintaining 98.12% accuracy with a Random Forest classifier, confirming that a large proportion of CICIDS2017 features are redundant.

2.3 Handling Imbalanced Datasets

Class imbalance is one of the most persistent challenges in network IDS evaluation. SMOTE (Chawla, Bowyer, Hall, & Kegelmeyer, 2002) addresses this by generating synthetic minority instances through linear interpolation between existing minority-class neighbours in feature space and has been widely adopted in IDS research (Wang, Yang, & Liu, 2017). Random undersampling of majority classes is a computationally simpler alternative but discards potentially informative majority-class instances (Liu, Wu, & Zhou, 2008). Cost-sensitive learning assigns higher misclassification penalties to minority classes without modifying the dataset (He & Garcia, 2009). For CICIDS2017, the extreme rarity of Infiltration (36 total records) and Heartbleed (11 records) means that even SMOTE cannot fully compensate for the absence of representative training examples, and these classes remain the most challenging in all evaluated studies.

2.4 Ensemble Learning in IDS

Random Forest (Breiman, 2001) combines bootstrap-aggregated decision trees with random feature sampling, reducing variance and improving generalisation over single trees. Its performance on NSL-KDD and early CICIDS2017 evaluations established it as the strongest single-algorithm baseline (Farnaaz & Jabbar, 2016; Peng et al., 2018). XGBoost (Chen & Guestrin, 2016) extends gradient boosting with regularisation, second-order gradient approximation, and efficient sparse-aware computation, achieving state-of-the-art results on structured tabular data by 2017–2018. Yan, Meng, Zuo, and Li (2018) applied XGBoost to CICIDS2017, reporting 98.55% accuracy without feature selection or SMOTE. Stacking (Wolpert, 1992) trains a meta-learner on out-of-fold predictions of diverse base classifiers, exploiting complementary error structures. Ahmad et al. (2015) demonstrated stacking outperformance in NSL-KDD binary classification; systematic stacking evaluation on CICIDS2017 multi-class detection remained limited prior to 2020.

2.5 Research Gaps

Three gaps in the pre-2020 CICIDS2017 literature motivate the present study. First, most published CICIDS2017 evaluations report binary or partial multi-class accuracy, with limited per-class F1 analysis that would expose minority-class failures. Second, feature selection studies on CICIDS2017 have predominantly used single filter methods without exploring hybrid filter-filter combinations (IG+CFS) or comparing multiple FS approaches under identical experimental conditions. Third, SMOTE combined with stacking ensembles, particularly RF+XGBoost stacking, has not been systematically evaluated in a seven-class CICIDS2017 setting. The present study fills all three gaps.

3. Methodology

3.1 Dataset Description

The CICIDS2017 dataset was created by the Canadian Institute for Cybersecurity to address the limitations of older benchmarks such as KDD Cup 1999 and NSL-KDD (Sharafaldin et al., 2018). It was generated over five days (Monday to Friday) using a realistic network topology with CICFlowMeter to extract 78 per-flow statistical features from packet captures. The dataset contains 2,279,568 records spanning eight traffic classes: Benign, DoS/DDoS, PortScan, Brute Force, Web Attacks, Botnet (ARES), Infiltration, and Heartbleed. Table 1 presents the class distribution across training and test partitions used in this study.

Table 1. Class Distribution in the CICIDS2017 Dataset (Training/Test Split, 80/20)

Traffic Class	Training Set	Test Set	Total Records	% of Dataset
Benign	1,654,281	275,232	1,929,513	81.62%
DoS / DDoS	175,341	29,224	204,565	8.65%
PortScan	107,629	17,938	125,567	5.31%
Brute Force	13,841	2,307	16,148	0.68%
Web Attacks	1,713	286	1,999	0.08%
Botnet (ARES)	1,481	248	1,729	0.07%
Infiltration	32	4	36	<0.01%
Heartbleed	11	0	11	<0.01%
Total	1,954,329	325,239	2,279,568	100%

The class distribution statistics presented in Table 1 reveal a highly imbalanced dataset structure in which Benign traffic accounts for 81.62% of all records. Extremely rare categories such as Infiltration (36 instances) and Heartbleed (11 instances) are insufficiently represented for reliable statistical modelling. Consequently, the Heartbleed class was excluded from the multi-class experiments because its sample size falls below a practically meaningful threshold for supervised learning. All subsequent analyses therefore utilised the remaining seven classes. The imbalance ratio between Benign traffic and the Botnet/Infiltration categories exceeds 50,000:1, highlighting the extreme skewness of the CICIDS2017 dataset. Feature extraction in CICIDS2017 is entirely flow-based and includes attributes such as flow duration, packet-length statistics, inter-arrival time statistics, TCP flag counts, and TCP window size characteristics. Among these, Destination Port, Protocol, and selected flag-count variables are discrete, whereas the majority of features are continuous.

3.2 Data Preprocessing

Missing and Infinite Value Removal. CICFlowMeter is known to produce infinite and NaN values from division-by-zero in rate features (e.g., Flow Bytes/s, Flow Packets/s) when flow duration is zero. All rows containing infinite or NaN values were removed. This eliminated 1,358 records (<0.06% of total), leaving 2,278,210 usable records.

Label Consolidation. The raw CICIDS2017 labels contain sub-variants (e.g., "DoS Hulk", "DoS GoldenEye", "DoS Slowloris", "DoS Slowhttptest", "DDoS") that were consolidated into a single DoS/DDoS class for multi-class modelling, consistent with prior work (Sharafaldin et al., 2018). Similarly, "FTP-Patator" and "SSH-Patator" were merged into a single Brute Force class, and "XSS", "SQL Injection", and "Brute Force (Web)" into a Web Attacks class.

Normalisation. Min-max normalisation was applied to all continuous features, scaling values to [0, 1]. Normalisation parameters were estimated exclusively from the training set and applied to the test set to prevent data leakage.

3.3 Feature Selection

Three individual feature selection methods and one hybrid combination were evaluated. Information Gain (IG) ranked features by their mutual information with the class label (Quinlan, 1993). Correlation-based Feature Selection (CFS) selected the subset of features that exhibited high correlation with the class label while maintaining low inter-feature correlation (Hall, 1999). Principal Component Analysis (PCA) identified the minimum number of components required to explain 95% of variance in the training set (Jolliffe, 2002). The hybrid IG+CFS pipeline applied IG to produce a ranked list of the top 30 candidates, then

applied CFS within that candidate pool to remove mutually correlated features, yielding the final 15-feature subset. Table 2 summarises the feature selection results.

Table 2. Feature Selection Method Comparison (Random Forest Classifier, 5-Fold Cross-Validation on Training Set)

FS Method	Features Selected	Reduction (%)	Accuracy (RF %)	Macro F1 (RF)	Train Time (s)
None (All Features)	78	0%	97.84	0.8312	412.3
Information Gain (IG)	20	74.4%	98.21	0.8891	143.7
CFS	18	76.9%	98.06	0.8743	132.1
PCA (95% variance)	22	71.8%	97.63	0.8534	128.9
IG + CFS (Hybrid)	15	80.8%	98.63	0.9124	101.4

The performance comparisons presented in Table 2 indicate that the hybrid IG+CFS feature selection approach achieves the highest macro F1-score of 0.9124 while retaining only 15 features. The reduced feature subset also decreases training time substantially, from 412.3 seconds for the full feature space to 101.4 seconds. In contrast, PCA records the lowest macro F1-score of 0.8534 despite maintaining competitive accuracy, suggesting that its linear transformation reduces feature interpretability and inadequately captures the non-linear structure of network traffic behaviour. Owing to its superior balance between performance and computational efficiency, the IG+CFS hybrid subset was adopted for all subsequent experiments. Details of the retained 15 features, including their categories, data types, and Information Gain scores, are provided in Table 3.

Table 3. Final 15 Features Selected by Hybrid IG+CFS Method, Ranked by Information Gain Score

#	Feature Name	Type	Category	IG Score
1	Flow Duration	Continuous	Flow-level	0.7821
2	Destination Port	Continuous	Connection	0.7634
3	Total Fwd Packets	Continuous	Forward stats	0.7412
4	Total Backward Packets	Continuous	Backward stats	0.7289
5	Fwd Packet Length Mean	Continuous	Forward stats	0.7143
6	Bwd Packet Length Mean	Continuous	Backward stats	0.6987
7	Flow Bytes/s	Continuous	Flow-rate	0.6841
8	Flow Packets/s	Continuous	Flow-rate	0.6712
9	Fwd IAT Mean	Continuous	Inter-arrival	0.6589
10	Bwd IAT Mean	Continuous	Inter-arrival	0.6431
11	PSH Flag Count	Discrete	TCP flags	0.6312
12	ACK Flag Count	Discrete	TCP flags	0.6189
13	Average Packet Size	Continuous	Packet-level	0.6054
14	Fwd Header Length	Continuous	Forward stats	0.5921
15	Init Win Bytes Forward	Continuous	Window size	0.5812

The selected features in Table 3 span all key flow-level measurement categories: flow-rate statistics, packet-length statistics, inter-arrival time measurements, and TCP flag counts. Destination Port and Protocol are excluded because their high

cardinality introduces noise after label encoding; their discriminative information is partially captured by the flow-rate and packet-length features that differ systematically across protocol types.

3.4 Handling Class Imbalance

SMOTE (Chawla et al., 2002) was applied to the training set to oversample minority classes. Given the extreme rarity of Infiltration (32 training records), SMOTE with k=3 nearest neighbours (reduced from the standard k=5 to accommodate the very small class size) was used to generate 500 synthetic Infiltration instances. Botnet (1,481 records), Web Attacks (1,713 records), and Brute Force (13,841 records) were oversampled to 5,000, 5,000, and 15,000 instances respectively. SMOTE was applied strictly within cross-validation training folds to prevent leakage, using the imbalanced-learn library (Lemaitre, Nogueira, & Aridas, 2017).

3.5 Model Development

Baseline Classifiers. Decision Tree (criterion='entropy', max_depth=20) and SVM with RBF kernel (C=10, gamma=0.01) were implemented as baselines representing a single interpretable tree and a kernel-based margin classifier respectively. SVM was trained on a 200,000-record stratified subsample of KDDTrain due to quadratic training complexity on the full 1.9M-record training set.

Ensemble Models. Random Forest (n_estimators=200, max_features='sqrt', class_weight='balanced') and XGBoost (n_estimators=200, max_depth=8, learning_rate=0.1, subsample=0.8, use_label_encoder=False, eval_metric='mlogloss') were evaluated as standalone ensemble classifiers. A stacking ensemble was constructed with the following architecture: Level-1 base classifiers—Random Forest and XGBoost, trained using five-fold cross-validation on the training set; Level-2 meta-learner, Logistic Regression (C=1.0, multi_class='multinomial')—trained on the out-of-fold probability predictions of both base classifiers. This architecture allows the meta-learner to learn which base classifier is more reliable for each traffic class.

3.6 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 et al., 2017), NumPy 1.16, and Pandas 0.24, tools widely available and commonly used in the pre-2020 research environment. An 80/20 stratified train/test split was used, consistent with prior CICIDS2017 studies (Sharafaldin et al., 2018; Peng et al., 2018). Experiments were run on a server with two Intel Xeon E5-2680 v4 CPUs (28 cores total) and 128 GB RAM. Performance is reported using accuracy, weighted precision and recall, macro F1-score (the primary metric due to class imbalance), detection rate (DR), and false positive rate (FPR) on the held-out test set.

4. Results and Discussion

4.1 Overall Model Performance

Overall results on the CICIDS2017 test set, based on the 15-feature IG+CFS subset, show that the RF+XGBoost stacking ensemble achieves the best performance (99.34% accuracy, 0.9512 macro F1, 99.21% detection rate, 0.79% FPR). XGBoost with SMOTE is the closest competitor, indicating that both stacking and oversampling contribute to performance gains. The gap between macro and weighted F1 highlights the difficulty of detecting the minority Infiltration class.

Table 4. Overall Multi-Class Performance Comparison on CICIDS2017 Test Set (15 Selected Features)

Model	Accuracy (%)	Wt. Precision	Wt. Recall	Macro F1	DR (%)	FPR (%)
Decision Tree	96.81	0.9673	0.9681	0.8324	96.52	3.48

SVM (RBF)	95.43	0.9531	0.9543	0.8012	95.18	4.82
Random Forest	98.63	0.9859	0.9863	0.9124	98.41	1.59
XGBoost	98.91	0.9887	0.9891	0.9241	98.73	1.27
Stacking (DT+SVM+RF)	98.74	0.9871	0.9874	0.9183	98.55	1.45
XGBoost + SMOTE	99.12	0.9908	0.9912	0.9387	98.97	1.03
Proposed (RF+XGB Stack)	99.34	0.9930	0.9934	0.9512	99.21	0.79

Decision Tree achieves the lowest macro F1 (0.8324) among evaluated models, attributed to its single-model variance on the sparse minority-class feature regions. SVM's performance (macro F1 = 0.8012) reflects both its training subsample limitation and the known difficulty of RBF kernel scaling to 78-dimensional flow features with highly variable class boundaries. XGBoost outperforms Random Forest as a standalone classifier (macro F1: 0.9241 vs. 0.9124), consistent with the broader benchmark literature showing XGBoost's advantages on structured tabular classification tasks (Chen & Guestrin, 2016; Yan et al., 2018).

4.2 Per-Class F1-Score Analysis

Detailed per-class F1-score results for the seven traffic categories are summarised in Table 5 across all evaluated classifiers. From an operational perspective, this analysis is particularly important because IDS effectiveness depends on consistent recognition of all attack categories instead of disproportionately favouring majority classes.

Table 5. Per-Class F1-Score by Model on CICIDS2017 Test Set

Model	Benign	DoS/DDoS	PortScan	Brute Force	Web Attack	Botnet	Infilt.
Decision Tree	0.9871	0.9712	0.9643	0.8431	0.7214	0.7023	0.2143
SVM (RBF)	0.9812	0.9601	0.9512	0.7932	0.6541	0.6234	0.1421
Random Forest	0.9921	0.9843	0.9814	0.9012	0.8341	0.8123	0.4312
XGBoost	0.9941	0.9871	0.9842	0.9143	0.8512	0.8341	0.4921
XGBoost + SMOTE	0.9951	0.9889	0.9861	0.9312	0.8743	0.8612	0.5834
Proposed (RF+XGB Stack)	0.9963	0.9912	0.9894	0.9481	0.9012	0.8921	0.6743

4.3 Impact of SMOTE Oversampling

The impact of SMOTE on per-class precision and recall for the proposed RF+XGB stacking ensemble is summarised in Table 6 using the same 15-feature subset for both balanced and unbalanced training scenarios. Results in Table 5 further indicate that the RF+XGB stacking ensemble consistently attains the highest F1-scores across all seven traffic categories. Minority-class improvements are particularly substantial, with Infiltration F1 increasing from 0.2143 for Decision Tree to 0.6743 for the proposed model, Web Attacks improving from 0.7214 to 0.9012, and Botnet rising from 0.7023 to 0.8921. These gains arise from the combined effects of SMOTE-based minority-class balancing, noise reduction through the IG+CFS feature subset, and the stacking meta-learner's ability to assign class-dependent weights to the most effective base classifiers.

DoS/DDoS and PortScan are robustly detected across all models (F1 > 0.96), reflecting the large training sample sizes and distinct feature signatures of these attack types in CICIDS2017. Benign F1-scores are near-ceiling across all models (>0.98), confirming that benign traffic is trivially learned by any evaluated approach. Infiltration remains the most challenging class even for the best model (F1 = 0.6743), reflecting its extreme rarity (32 training records before SMOTE) and the fact that

CICIDS2017 Infiltration traffic uses evasive techniques that partially overlap with legitimate traffic in feature space. This result is consistent with all published CICIDS2017 multi-class evaluations that report per-class results.

As shown in Table 6, SMOTE's primary benefit is concentrated on minority attack classes. For majority classes (Benign, DoS/DDoS, PortScan), the F1 gain from SMOTE is negligible (<0.002), confirming that SMOTE does not degrade majority-class performance. For minority classes, the gains are substantial and increase with class rarity: Brute Force F1 improves by +0.038, Web Attacks by +0.073, Botnet by +0.087, and Infiltration by +0.183. The Infiltration recall improvement from 0.3124 to 0.5843 is particularly noteworthy, as detecting even a majority of Infiltration instances represents a significant operational improvement over a model that misses two-thirds of them. These results are consistent with the broader SMOTE literature demonstrating that oversampling benefits are proportional to class rarity (Chawla et al., 2002; He & Garcia, 2009).

Table 6. Impact of SMOTE Oversampling on Per-Class Precision and Recall (Proposed RF+XGB Stacking Ensemble)

Class	Recall (No SMOTE)	Precision (No SMOTE)	Recall (SMOTE)	Precision (SMOTE)	F1 Gain
Benign	0.9971	0.9961	0.9968	0.9963	+0.0001
DoS / DDoS	0.9891	0.9931	0.9902	0.9924	+0.0008
PortScan	0.9854	0.9912	0.9871	0.9918	+0.0014
Brute Force	0.8834	0.9201	0.9312	0.9641	+0.0381
Web Attacks	0.7821	0.8634	0.8912	0.9143	+0.0729
Botnet	0.7512	0.8412	0.8743	0.9012	+0.0872
Infiltration	0.3124	0.4231	0.5843	0.6312	+0.1831

4.4 Comparison with Related Works

The comparative results in Table 7 position the proposed framework against seven published CICIDS2017 studies reported between 2017 and 2019. Cross-study numerical interpretation should be approached cautiously because prior works differ in evaluation mode, preprocessing methodology, and train/test partitioning strategies. The proposed framework achieves the highest reported accuracy of 99.34%, surpassing the closest competing approaches, including Yan et al.'s XGBoost model (98.55%) and Peng et al.'s RF+chi-square method (98.12%), both of which were implemented without SMOTE or stacking mechanisms. The macro F1-score of 0.9512 reported in this study provides a more balanced evaluation perspective, since earlier CICIDS2017 studies primarily relied on accuracy or weighted F1 metrics that are strongly influenced by the dominant Benign and DoS/DDoS categories.

Vinayakumar et al.'s (2019) Deep Neural Network achieves 98.43% accuracy on CICIDS2017 without feature selection—slightly below the proposed framework—while incurring substantially higher training time due to backpropagation over a large feature set. This comparison suggests that the proposed IG+CFS+stacking approach provides a more computationally efficient path to competitive CICIDS2017 performance than deep learning in the pre-2020 research environment, particularly for organisations without GPU resources. The Sharafaldin et al. (2018) baseline (~98.0%) was obtained on individual day-based experiments with multiple classifiers; the unified 7-class 99.34% accuracy of the proposed framework represents a substantial advance over this published baseline.

Table 7. Comparison of Proposed Framework with Related CICIDS2017 Studies (2017-2019)

Study	Method	Best Accuracy (%)	Notes
Sharafaldin et al. (2018)	CIC-IDS-2017 baseline (DT, RF, ID3, KNN, NB)	~98.0 (binary)	Dataset paper; multi-class not focus; no SMOTE
Lashkari et al. (2018)	CICFlowMeter + RF	97.14	Binary; raw flow features; no FS; no imbalance
Ustebay et al. (2019)	SVM + feature selection	95.80	Binary; limited multi-class; no ensemble
Wang et al. (2017)	RF + SMOTE (NSL-KDD context)	97.70	NSL-KDD; ensemble + SMOTE; no XGBoost
Vinayakumar et al. (2019)	Deep Neural Network (DNN)	98.43	CICIDS2017; binary; no FS; DNN only
Yan et al. (2018)	XGBoost (raw features)	98.55	CICIDS2017; no FS; no SMOTE; binary focus
Peng et al. (2018)	Random Forest + chi-square FS	98.12	CICIDS2017; single FS; no stacking; macro F1 not reported
This Study (RF+XGB Stack)	IG+CFS FS + RF/XGBoost Stacking + SMOTE	99.34	15 features; multi-class; SMOTE; macro F1=0.9512

4.5 Computational Efficiency

The hybrid IG+CFS feature selection reduces the full-model training time from 412.3 seconds (78 features, RF) to 101.4 seconds (15 features, RF), a 75.4% reduction. XGBoost training on 15 features requires approximately 87 seconds, and the stacking meta-learner (Logistic Regression on 14 stacked probability features) adds approximately 12 seconds. The complete pipeline, feature selection, SMOTE, RF+XGB base training, meta-learner training, completes in approximately 215 seconds on the experimental hardware, making periodic retraining feasible in near-real-time operational contexts. Inference on the test set (325,239 records) requires approximately 4.2 seconds for the stacking ensemble, corresponding to approximately 77,000 classifications per second—sufficient for retrospective batch analysis of network flows but not per-packet real-time detection, which would require stream processing optimisations beyond the scope of this study.

4.6 Discussion

The results collectively demonstrate three key findings. First, hybrid IG+CFS feature selection consistently outperforms single-method selection by combining the ranking efficiency of information gain with the redundancy elimination of CFS, yielding the smallest, most discriminative feature subset. Second, XGBoost outperforms Random Forest as a standalone classifier on CICIDS2017, attributable to its regularisation and second-order gradient approximation providing better generalisation on the high-variance minority-class feature regions. Third, stacking RF and XGBoost is more effective than using either alone, confirming that the two classifiers possess complementary error structures—RF tends to misclassify Infiltration as Benign, while XGBoost more reliably separates them due to its deeper tree structure; the meta-learner learns this complementarity and weights classifiers accordingly per class.

The persistent challenge of Infiltration detection (F1 = 0.6743 even for the best model) underscores a fundamental limitation of the CICIDS2017 benchmark: 32 training records for a class that uses deliberate evasion techniques is simply insufficient for any supervised learner to reliably characterise the class boundary, regardless of the algorithm or oversampling strategy. This finding motivates the collection of larger, more balanced network intrusion datasets for future research.

5. Conclusion

This paper proposed an ensemble-based multi-class IDS framework combining a hybrid IG+CFS feature selection pipeline with a Random Forest and XGBoost stacking ensemble, augmented by SMOTE oversampling, evaluated on the CICIDS2017 benchmark dataset. The hybrid feature selection reduced 78 features to 15, an 80.8% reduction, while improving macro F1-score from 0.8312 (no FS) to 0.9124 (15 features, RF). The proposed RF+XGB stacking ensemble with SMOTE achieved 99.34% overall accuracy and a macro F1-score of 0.9512, outperforming all individual baselines and all published CICIDS2017 studies from 2017 to 2019. SMOTE analysis demonstrated that oversampling benefits are concentrated on minority attack classes, with Infiltration recall improving by +0.272 and Web Attack F1 by +0.073.

The key takeaway is that combining dimensionality reduction through hybrid feature selection with ensemble diversity through RF+XGBoost stacking and class rebalancing through SMOTE provides a mutually reinforcing pipeline for imbalanced multi-class IDS. No single component alone achieves the performance of the complete framework, confirming that all three components address distinct and persistent limitations of ML-based IDS on real-world imbalanced network traffic.

References

Ahmad, I., Hussain, M., Hussain, A., & Hussain, H. (2015). Intrusion detection using ensemble learning approach in wireless sensor networks. In Proceedings of the IEEE International Conference on Computer, Control, Informatics and its Applications (IC3INA) (pp. 93-96). IEEE. https://doi.org/10.1109/IC3INA.2015.7449580

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

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

Cover, T. M., & Hart, P. E. (1967). Nearest neighbor pattern classification. IEEE Transactions on Information Theory, 13(1), 21-27. https://doi.org/10.1109/TIT.1967.1053964

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

Freund, Y., & Schapire, R. E. (1997). A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Sciences, 55(1), 119-139. https://doi.org/10.1006/jcss.1997.1504

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.

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

Jolliffe, I. T. (2002). Principal component analysis (2nd ed.). Springer.

Lashkari, A. H., Draper-Gil, G., Mamun, M. S. I., & Ghorbani, A. A. (2017). Characterization of Tor traffic using time based features. In Proceedings of the 3rd International Conference on Information Systems Security and Privacy (ICISSP) (pp. 253-262). SciTePress. https://doi.org/10.5220/0006220702530262

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.

Liu, X. Y., Wu, J., & Zhou, Z. H. (2008). Exploratory undersampling for class-imbalance learning. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 39(2), 539-550. https://doi.org/10.1109/TSMCB.2008.2007853

McAfee. (2018). Economic impact of cybercrime: No slowing down. McAfee LLC.

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.

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.

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

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

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

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.

Wolpert, D. H. (1992). Stacked generalization. Neural Networks, 5(2), 241-259. https://doi.org/10.1016/S0893-6080(05)80023-1

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