Middle East Journal of Humanities and Cultural Studies
Homepage: http://meijournals.com/ar/index.php/mejljs/index
ISSN: 2710-2238 (PRINT)
ISSN: 2788-4686 (ONLINE)
مجلة الشرق الأوسط
للعلوم الإنسانية والثقافية

Assessment and mitigation of DDOS attacks in large scale networks

Mohammed Wael Rasheed ALLQASAM
AL-Nahrain University, Computer Center, Baghdad, Iraq
Mohammedwr90@Gmail.com

استلام البحث:14-06-2026 مراجعة البحث :13-07-2026 قبول البحث:09-08-2026

Abstract:
Distributed Denial-of-Service (DDoS) attacks are a serious problem in today's networked world, especially with the rise of Internet of Things (IoT), Software-Defined Networking (SDN), cloud computing and 5G deployment that are driving the proliferation of network traffic in scale and complexity. This study suggests an integrated machine-learning framework for detecting multiclass DDoS attacks and mitigating the attacks by simulation; especially significant issues addressed in this study are severe class imbalance, redundant features, and reliable recognition of minority classes. The proposed framework is tested with the CIC-IDS2017 benchmark dataset using two stages experimental design. In Phase 1, a conventional baseline is created with three algorithms: Random Forest (RF), XGBoost (XGB) and LightGBM (LGB) with adaptive SMOTE. Rare-class filtering, training-only feature selection, SMOTE, class-weighted learning, and soft-voting ensemble classification are introduced in phase 2. A reproducible 20% sample of CIC-IDS2017 is used followed by an 80:20 stratified train-test split, and all the learned preprocessing processes are applied to the training data only to avoid information leakage. This feature selection further shrinks the input space from 77 to 30 features that are the most informative. Results show that the most accurate model is Random Forest with an accuracy of 99.83% and a Macro F1-Score of only 0.7353 indicating the shortcomings of accuracy while working with severe class imbalance. By contrast, the proposed Phase 2 ensemble achieves 99.51% accuracy, 0.8108 Macro Precision, 0.9409 Macro Recall, 0.8509 Macro F1-Score and 0.9956 Weighted F1-Score. Therefore, Macro F1 is enhanced by 0.1156 compared to the baseline and the Macro Recall is significantly enhanced. The integrated framework also consists of an integrated confidence-based mitigation simulation that blocks 72.23% of the prespecified synthetic attack scenario. The results indicate that this preprocessing method along with supervised feature reduction, weighted learning and probabilistic ensemble classification yields better balanced multiclass DDoS detection than the conventional methods that focus on accuracy.
Keywords: Assessment, mitigation, DDOS attacks, large scale networks

1. Introduction:

Distributed Denial-of-Service (DDoS) attacks are a growing, pervasive and escalating type of attack that impacts the availability and reliability of modern networked systems. Attackers can cause overwhelming bandwidth, computing power or network services by having many compromised and malicious sources flooding the network, thereby interrupting legitimate users. With the advent of new technologies such as IoT, Software-Defined Networking (SDN), cloud computing and 5G technology, the attack surface area of DDoS attacks has been expanding and growing more complex (Tiwana et al., 2024; Djuitcheu et al., 2023; Singh & Jain, 2024).

This is also highlighted by recent large-scale observations, which show that this is becoming a more important threat. Cloudflare said it thwarted around 14 million DDoS attacks in 2023 and 21.3 million in 2024, a rise of 53%. The number of attacks more than doubled again in 2025 to 47.1 million, an increase of 121% over 2024. Overall, there was a rise in attacks of about 236% between 2023 and 2025. Figure 1 shows a remarkable surge in the volume of DDoS attacks from 2023 until 2025.The figure 1 illustrates the remarkable surge that has

occurred in the volume of DDoS attack from 2023 till 2025. Number of DDoS attacks mitigated by Cloudflare, illustrating the rapid growth of the threat over recent years.

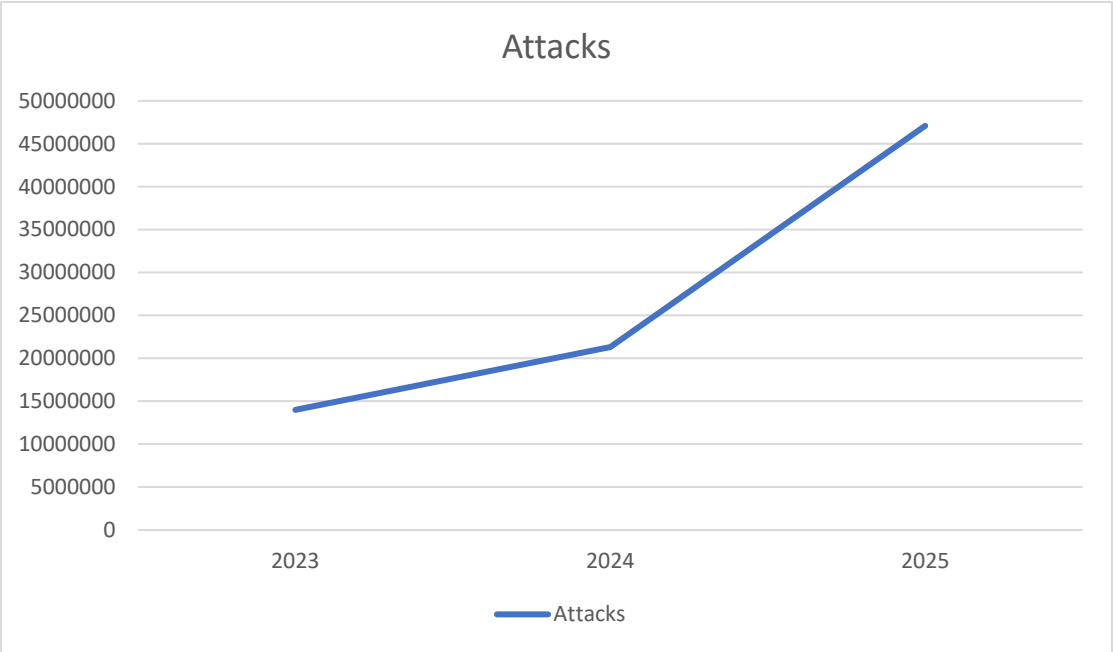

Figure 1. Growth in the number of DDoS attacks mitigated by Cloudflare, 2023–2025.

Source: Cloudflare DDoS Threat Reports, 2024–2025.

The escalating incidence of DDoS shows that it's not only a persistent issue, but also growing in size, as shown in Figure 1. As this growth continues, there is a greater demand to be able to detect a wide range of attack patterns quickly and accurately, and take some action to mitigate the impact of the attack prior to a significant effect on network availability. The challenge is even more significant due to the widespread nature of modern DDoS attacks, which can range in traffic characteristics, intensity, duration and attack vectors. For instance, Cloudflare said there were over 21.3 million attacks successfully mitigated in 2024, with the biggest that they saw coming in at 5.6 Tbps.

Signature, threshold, or traffic type detection methods are well suited to defend against attacks they have already been exposed to, but may not be as effective if an attack changes its behavior or appears similar to legitimate network traffic. Therefore, machine-learning methods have been looked at for the detection of DDoS attacks, since they can learn intricate relationships between network-flow features. Ensemble voting, Random Forest, Decision Tree, entropy-based learning, genetic algorithms, federated learning, and feature-optimized machine learning approaches are some recent studies that have been undertaken (Alsumaidaie et al., 2024; Hassan et al., 2024; Saiyed & Al-Anbagi, 2024; Fotse et al., 2024; Ibrahim et al., 2025).

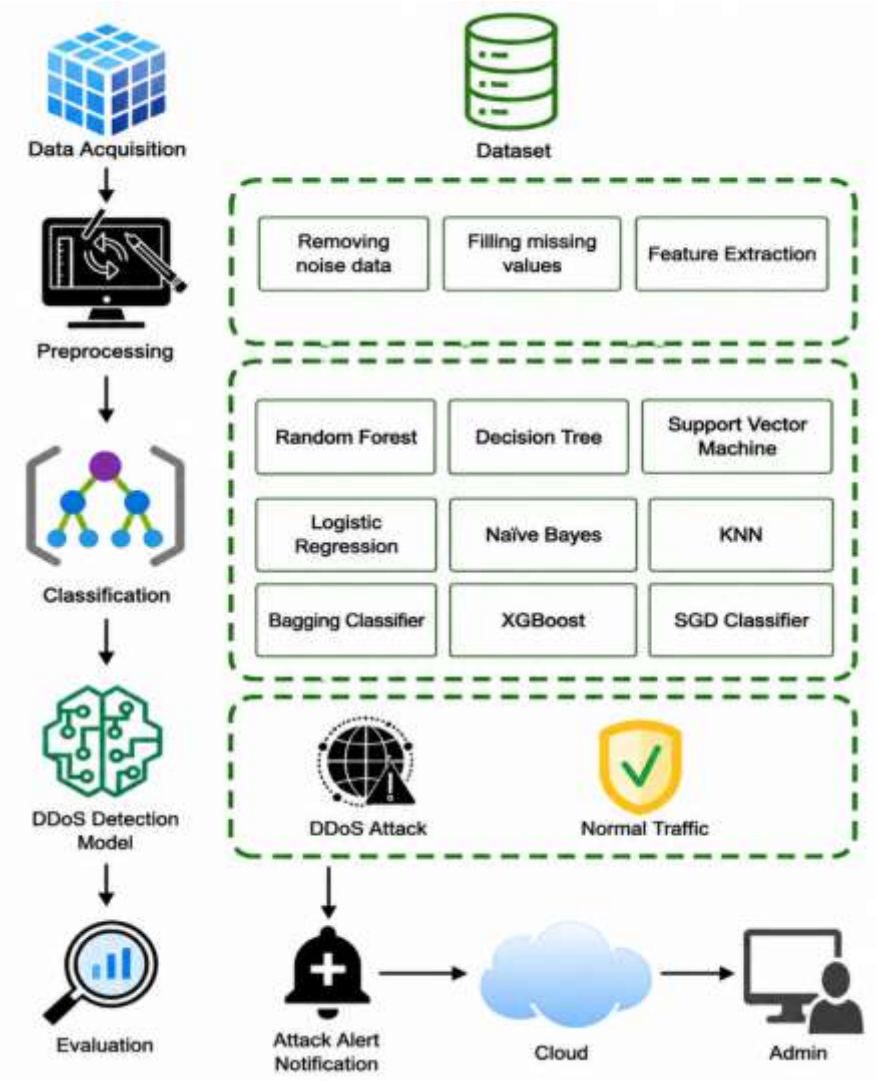

Figure 2. General architecture of machine-learning-based DDoS detection and mitigation.

Intelligent DDoS defense usually consists of traffic collection, preprocessing, feature selection, classification and mitigation as described in Figure 2. This process has been furthered by distributed and adaptive techniques such as blockchain-SDN with neural networks (Tiwana et al., 2024), federated learning (Fotse et al., 2024), entropy-based detection (Hassan et al., 2024), feature optimization (Ibrahim et al., 2025), and genetic-algorithm-based feature selection (Saiyed & Al-Anbagi, 2024). However, there are significant challenges, especially with respect to class imbalance, redundant feature sets, computational complexity and weak mitigation. A high overall accuracy can also mask low accuracy for minority classes. Therefore, a methodological gap still exists in designing a simple, reproducible, leakage free and imbalance aware multiclass framework that effectively represents features and also effectively classifies using ensemble learning. In order to fill this gap, the present study proposes an integrated CIC-IDS2017-based framework which includes stratified sampling, data cleaning, feature selection, training-only adaptive SMOTE, class weighting and independent Random Forest, XGBoost and LightGBM classifiers with soft voting ensemble learning. The framework focuses on Macro Precision, Macro Recall, Macro F1-Score and per-class performance, with a simulation-based mitigation component assessing the feasibility of the connection between detection and automated response.

I. Literature Review

Machine learning (ML) and deep learning (DL) have been applied for DDoS detection and detection systems that use these technologies have grown in number as they are able to detect and capture complex traffic patterns that are not detectable by traditional rule-based detection methods. In recent studies, various approaches with methods have been used. Alshdadi et al. (2024) proposed EffiGRU-GhostNet, an ensemble of deep-learning models based on GhostNet and GRU architectures with the combination of PCA for dimensionality reduction to obtain complementary traffic representation. The approach obtained an accuracy of 97.88% for IoT-23 and 99.99% for APA-DDoS. To prevent DDoS attacks in the vehicular network, Verma et al. (2024) proposed PREVIR, which integrates statistical logit analysis with LogitBoost to attain up to 99.99% accuracy and 100% TPR. Varalakshmi and Thenmozmi (2025) proposed an entropy-based stochastic detection and mitigation mechanism for SDN-IoT networks to detect and mitigate such attacks as TCP, UDP and ICMP SYN-flood using entropy characteristics. To mitigate the limitation of CNN-based feature extraction and LSTM-based sequential learning in DDoS classification, Danang and Mustofa (2026) presented a novel CLSTMNet architecture which is a hybrid of CNN and LSTM.

The reported accuracy is generally high, although the previous studies have different datasets, preprocessing procedure, and evaluation setting, thus, it is hard to compare the studies directly as showing in Figure 3. Moreover, accuracy alone may not adequately reflect minority-class recognition in imbalanced datasets. Therefore, the present study focuses on multiclass detection, macro-averaged metrics, feature selection, and explicit class-imbalance handling to provide a more comprehensive evaluation.

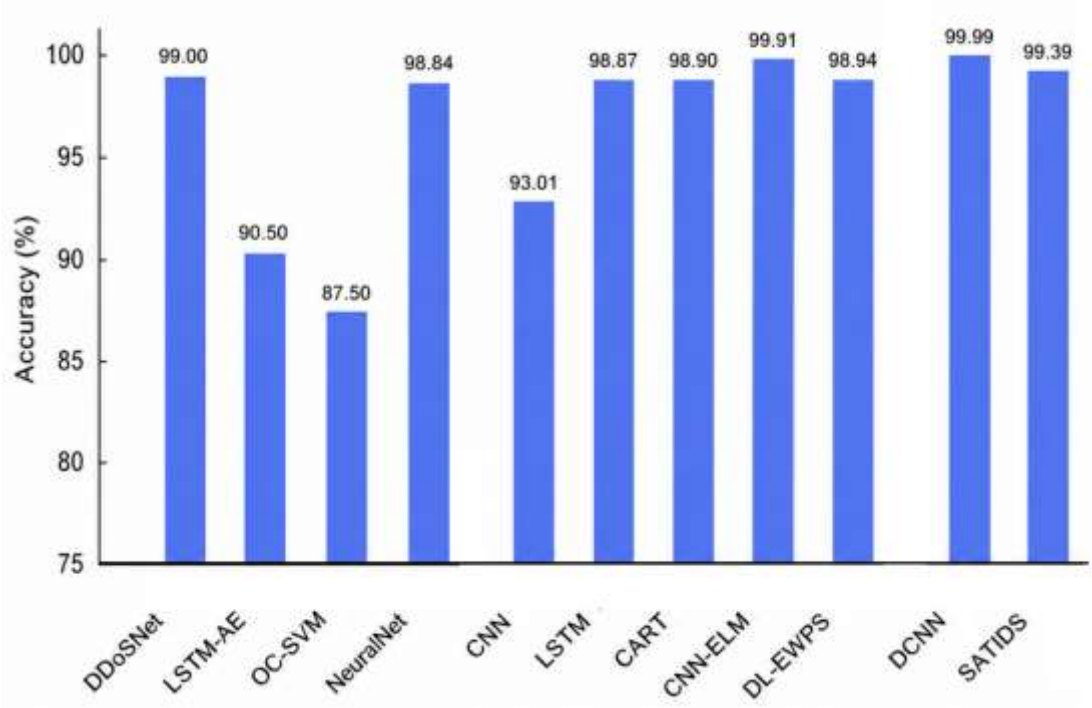

Figure 3. Reported DDoS detection accuracy according to (Clinton., et al, 2024).

Although high benchmark accuracy is often mentioned, recent research shows that the transferability of DDoS detection models to new traffic environments is still a big challenge. To address the problem, Khalil et al. (2025) explored the CNN-based models trained with the benchmark traffic and then tested them with the new 5G traffic on a simulated 5G Multi-Access Edge Computing (MEC) network. For CICDDoS2019-UDP, LUCID CNN achieved an F1 score of 0.9686 with 96.98% validation accuracy and Mohak CNN achieved an F1 score of 0.9470 with 94.99% validation accuracy. When the models were applied to new simulated data, however, they performed poorly. In contrast, LUCID achieved an F1-score of zero for both BoNeSi+Simu5G and Mixed Simu5G, while Mohak only reached an F1-score of 0.0055 for BoNeSi+Simu5G and 0.8976 for Mixed Simu5G. These results are particularly significant as being evidence that good benchmark performance does not necessarily translate to good generalization to new traffic distributions. Khalil et al. (2025) report the cross-dataset results, as shown in Figure 4.

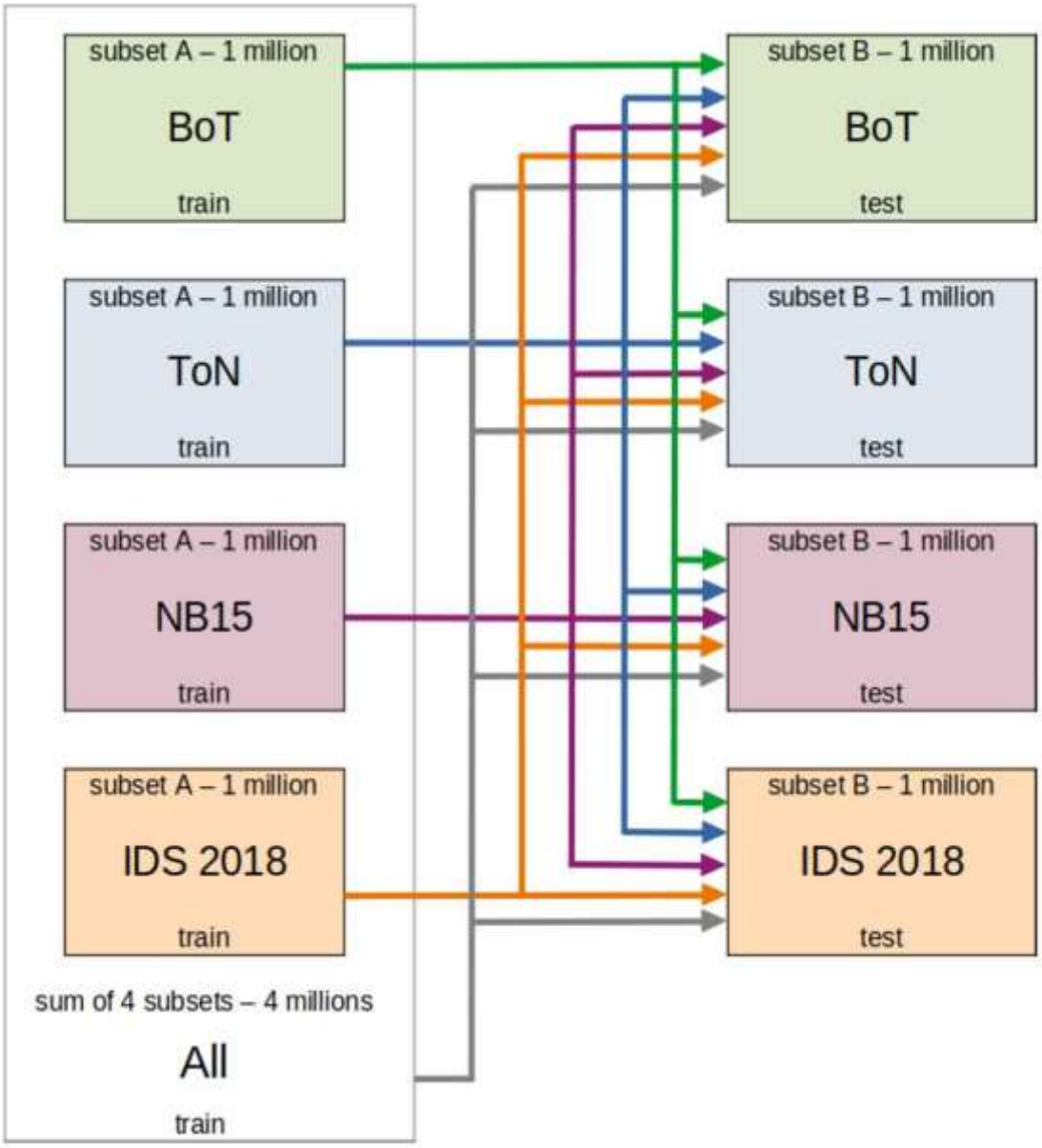

Figure 4. Cross-dataset generalization of DDoS detection models reported by Khalil et al. (2025).

Figure 4 demonstrates that there are significant differences in performance when comparing the benchmark with the newly generated traffic, emphasising the challenges of dataset diversity and model generalization. Therefore, Khalil et al. (2025) advocated for both transferability and deployment feasibility, in addition to accuracy. Likewise, Singh et al. (2022) showed that there are 13 vulnerabilities in the ONOS SDN controller, highlighting the need for additional consideration of the security of the infrastructure containing the network controller in addition to traffic-level detection. Olufemi et al. (2025 also pointed out the SDN controllers, OpenFlow, network slicing, edge computing, and 5G IoT environments as vulnerable areas, emphasizing the need for adaptive security mechanisms. Recent research has also evolved from the detection to automated mitigation and recovery. Selective ML-based mitigation achieved better detection accuracy (14%) and a throughput improvement of over

440% according to Joo et al. (2025). In this context, Köksal et al. (2024) presented a lightweight mitigation framework for MEC networks based on the Kubernetes framework, while Alrumaih and Alenazi (2025) recovered around 88% of the normal throughput with adaptive recovery. This was also shown by Qin et al. (2025) and Verma et al. (2024) who reported on the advantages of automated traffic control and proactive DDoS prevention. Taken together, these studies suggest a shift from passive to closed-loop DDoS defense mechanisms (including detection, mitigation and recovery).

This progress has left out of course some gaps. First, numerous studies focus on the overall accuracy and lack of consideration for response accuracy of minority classes and serious class imbalance. Second, achieving a high benchmark score does not necessarily imply a high generalization score. Third, features and architectures which are redundant in terms of computing may reduce the deployment possibilities. Lastly, mitigation solutions are restricted to certain network types, e.g., MEC, WSN, vehicular or ICS networks, and hence can only be used in these specific types of networks. Therefore, there is a need for an integrated, leakages controlled and imbalanced aware multiclass framework, which includes effective feature representation, powerful ML classification, and automated mitigation.

In this regard, the present study proposes an integrated framework which is a combination of stratified sampling, data cleaning, feature selection, training only adaptive SMOTE, class weighting, and soft voting ensemble learning of independent Random Forest, XGBoost and LightGBM classifiers, respectively. The metrics include accuracy, Macro Precision, Macro Recall, Macro F1-Score, and per-class metrics, as well as a simulation-based mitigation component. The methodology proposed and the experimental process are described in the following section.

II. Methodology

The proposed methodological framework is built upon two phases of sequential pipelines to systematically tackle the critical challenges in detecting the DDoS attack in a large-scale network, which are severe class imbalance, high dimensional feature space and poor generalizability to rare attack patterns. The first phase sets a conventional baseline to uncover built-in weaknesses, and the second phase proposes a series of specific improvement measures to address limitations and leads to a realistic mitigation simulation.

2.1. Dataset and Preliminary Processing

The experiments are carried out on the CIC-IDS2017 benchmark dataset which includes benign network traffic and various attack categories such as DDoS, DoS, Bot, PortScan and Web Attack traffic. The dataset files are first combined into a single dataset. A random sample of 20% of the total traffic is taken, with a fixed random seed (42), to reduce the computation needs but still capture a wide range of the original traffic distribution. After all the cleaning and filtering, there are 566,149 observations in the resulting subset.

The initial pre-processing steps include the following:

1. Data Aggregation and Sampling: All the individual CIC-IDS2017 CSV files are merged into a single data set. To decrease the computational complexity, a 20%

random sample is extracted from this, but the sample is not reproducible if random seed 42 is used. Importantly, the 20% sampling operation is done prior to the next train-test partition, and it is not used as a replacement of the stratified train-test splitting.

2. Data cleaning: non-predictive identifiers such as Flow ID, Source IP, Destination IP, Timestamp, and other identifier fields as applicable are deleted. Infinite values are replaced by missing values and missing numbers are replaced by zero in order to ensure numerical stability when processing the data later.
3. Categorical Encoding: Categorical input features are encoded as numbers using label encoding, the output feature is encoded separately with a separate label encoding system. Target labels are deleted if empty or invalid prior to class filtering and model development.

There were 2,830,743 observations in the complete CIC-IDS2017 dataset created from the available traffic files. A random sample of 566,149 observations was then obtained, representing 20%.A 20% random sample was then drawn, giving 566,149 observations. Following exclusion of invalid or empty target labels, there were 566,138 observations used to develop the model. After the rare-class filtering, the resulting data set had twelve traffic classes. This sampling strategy was only used to limit computational costs, but class proportions within the sample set were maintained in the train test partition by stratified splitting.

2.2. Rare-Class Filtering

After sampling the dataset to be reproducible for 20%, the minimum-support criterion was applied to guard against noisy minority-class estimates, as a minority class may not be adequately represented by the observations in the database. Classifications of less than 100 observations in the sample set were not used in further model development. This threshold was chosen as a practical minimum support threshold, as it was important that each class retained, provided enough examples for both the training and testing partition in stratified splitting.

After this filtering operation, only twelve traffic classes were still left: BENIGN, Bot, DDoS, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, FTP-Patator, PortScan, SSH-Patator, Web Attack–Brute Force, and Web Attack–XSS. After filtering, there were 566,138 observations.

When using the filtering criterion, the filtering is only used to define the experimental class space and is not used to generate synthetic samples or to obtain information from the held-out test predictions.

The minimum-support threshold was introduced to separate theoretically possible classification from statistically sound supervised learning. Classes with fewer than 100 observations in the sampled data set perform poorly in terms of the amount of empirical evidence provided to learn stable class-specific decision boundaries especially after the subsequent train-test partition. These classes would yield very few trainings and testing

supports, and estimates of performance would be very sensitive to individual observations. This threshold was chosen as an experiment, not because these classes of attacks are necessarily of little consequence. The excluded categories are also valid for a cyber security point of view, but having been less represented in the sampled benchmark, there is a lack of reliability in the quantitative evaluation in this case, due to the limited number of samples in the present experimental setup.

2.3. Data Partitioning and Leakage Prevention

The dataset was then stratified and split into 80:20 training and testing sets, using a random seed of 42 for the filtering. The relative distribution of the retained traffic classes was preserved in both subsets by using stratification. The test subset was not used for model fitting, feature selection, synthetic sample generation or parameter estimation and was only used for final test performance evaluation.

All transformations of learned parameters were fitted on the training partition only, to minimize the leakage of information. The StandardScaler was trained on the training set of data, and then applied to the test set using the parameters learnt from the training part. Likewise, an XGBoost estimator was trained on just the training data for feature selection. SMOTE was performed after feature selection and only on the training subset. As a result, the data used for the test remained completely untouched from oversampling and model training.

2.4. Phase 1: Baseline Methodology (Conventional Approach)

The first phase sets a baseline of conventional machine learning to then be compared with the proposed methodology. Instead of applying rare-class filtering, feature selection, class weighting, or ensemble-based decision making, the baseline purposefully keeps all classes of traffic and the full range of features. This design will serve as a control to measure the impact of the improvements targeted in Phase 2.

- **Direct Imbalance Handling**

Following the stratified 80:20 train-test split, the complete training set is subjected directly to the Synthetic Minority Over-sampling Technique (SMOTE). At this point, no rare class filtering is done, which means that every class found in the sampled dataset is kept. To alleviate the level of class imbalance, SMOTE creates synthetic minority samples by smoothly interpolating between actual minority samples and their k nearest neighbors.

The CIC-IDS2017 dataset is not normally distributed, as the sample sizes of the classes are significantly different, therefore an adaptive neighborhood parameter is used to make SMOTE valid for the small minority classes. In particular, the number of nearest neighbors is defined as:

$$k = min(5, n_{min} - 1)$$

where n_{min} is the smallest number of training samples in any class. This adaptive configuration keeps the same baseline, SMOTE, and also avoids algorithm failures if there are less than 6 minority class samples in the training set. Importantly, SMOTE is used solely on the training set, and the held-out test set is not modified, so that it is evaluated in a fair and unbiased fashion.

- **Model Training**

The training data is resampled using the SMOTE method and three independent ensemble classifiers are trained on the resampled training data:

- Random Forest (RF): considered a strong baseline ensemble method that is able to model nonlinear relationships and shows relatively stable performance in a heterogeneous traffic pattern.
- XGBoost (XGB): used as a gradient boosting classifier for well-structured and tabular classification tasks.
- LightGBM (LGB): a framework for gradient boosting that is efficient for large-scale network-traffic data.

There is no voting nor stacking process in Phase 1. The classifiers each make their own prediction, so that the individual performance of the three conventional learners can be directly compared.

- **Evaluation Protocol**

The baseline models are tested on the unseen 20% test set which is not oversampled and not used for model fitting. The following accuracy, precision, recall, F1-Score, and Macro F1-Score are used for the assessment of performance. Special attention is paid to Macro F1-Score due to its equal weights for all traffic classes and being able to better inform the results than accuracy or weighted score in case of severe class imbalance. The configurations of the hyperparameters used by the three baseline classifiers are detailed in Table 2 and Figure 5.

Table 2. Hyperparameter Configuration for Baseline Models (Phase 1)

Model	Hyperparameter Configuration	Justification
Random Forest	Number of trees = **100**; random seed = **42**; parallel processing enabled	Provides a robust baseline while maintaining computational efficiency and reproducibility.
XGBoost	Number of boosting rounds = **150**; learning rate = **0.05**; maximum depth = **6**; subsample = **0.8**; column subsampling = **0.8**; evaluation metric = **mlogloss**; random seed = **42**	Provides a controlled gradient-boosting configuration for nonlinear multi-class intrusion classification.
LightGBM	Number of boosting rounds = **150**; learning	Provides an efficient gradient-

	rate = **0.05**; number of leaves = **31**; subsample = **0.8**; random seed = **42**	boosting baseline suitable for large-scale tabular network traffic data.

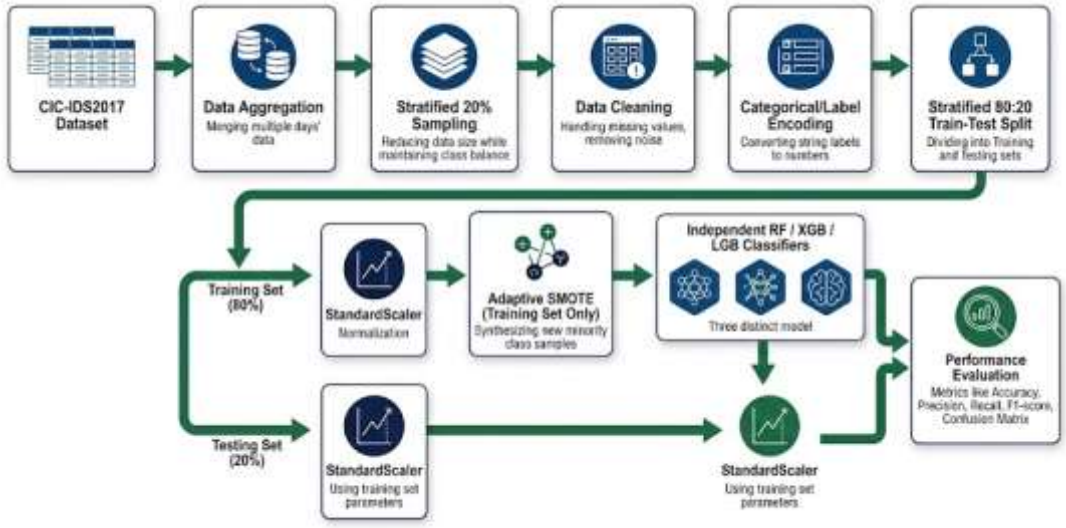

Figure 5: Conceptual Pipeline of Phase 1 (Baseline)

The baseline approach (Phase 1) is a traditional pipeline where all the features kept are fed directly to the classifiers without performing any transformation and the class imbalance is handled mainly by the use of adaptive SMOTE. The adaptive strategy is allowed to operate with very small minority classes but it does not explicitly exclude extremely rare classes, select the most discriminative features, or set class-specific training weights. In such cases, the baseline can therefore be susceptible to the presence of noisy or less dominating minority patterns and using the full feature space can add redundant and less informative dimensions. The constraints prompt the proposed design improvements in Phase 2 that include rare-class filtering, supervised feature selection, class-weight optimization and soft-voting ensemble learning.

2.5. Phase 2: Proposed Enhanced Methodology

The proposed methodology combines four controlled enhancements: rare-class filtering, training-only feature selection, SMOTE based oversampling, and weighted soft-voting ensemble learning. Two experimental phases are taken into consideration. All 15 traffic classes, ranging from extremely rare attack classes, are maintained after preliminary pre-processing, which is the standard learning scenario for Phase 1. There are 12 classes for Phase 2 due to fewer than the required minimum support of 100 observations following the reproducible sampling stage in three classes. Classes with sample sizes far too small to provide the observations needed to train and evaluate a reliable model, resulting in unstable learning and evaluation of the minority classes, and possibly unreliable estimates of the

Macro F1-Score, were excluded. Thus, the 12-class space represents a practically learnable class space, rather than an arbitrary reduction in the number of classes.

The proposed Phase 2 pipeline consists of:

1. Rare-Class Filtering: Three classes with fewer than 100 observations are removed, resulting in 12 classes.
2. Training-Only Feature Selection: XGBoost ranks features using only the training set, retaining the top 30 of 77 features.
3. SMOTE Resampling: Synthetic Minority samples are created only from selected training samples.

Weighted Ensemble Learning: XGBoost and LightGBM are trained with class weights, and the probability output of each are combined via soft voting.

Therefore, the limitation of traditional learning is illustrated in Phase 1 when the learning classes are very few, while in Phase 2 the proposed learning system is assessed once the learning space is sufficiently supported. The improvement is therefore evaluated not only on the basis of the accuracy but also on the basis of the multiple effects of rare class management, feature reduction, imbalance mitigation, and ensemble learning.

Table 2: Methodological Differences Between Phase 1 and Phase 2

Component	Phase 1 (Baseline)	Phase 2 (Proposed)
Rare Class Handling	None (Applied SMOTE only)	Rare Class Filtering (< 100 samples excluded)
Feature Space	All 77 features used	Top 30 Features selected
Class Weighting	Not applied	Inverse Frequency Weights applied during training
Decision Strategy	Individual Classifiers (RF, XGB, LGB)	Soft Voting Ensemble (XGB + LGB)
Final Output	Single model prediction	Aggregated probabilistic prediction

Ablation analysis is performed to quantify the single contribution of the main elements of the proposed framework by gradually adding the methodological improvements. Five configurations are tested: (A) baseline configuration with complete feature space and SMOTE; (B) baseline configuration with rare-class filtering; (C) filtered configuration with supervised feature selection; (D) filtered and feature-selected configuration with class-weighted learning; and (E) complete proposed framework with soft-voting ensemble learning. To ensure a controlled comparison, the same training-test partition, and the same random seed are used for all configurations. The results on accuracy, Macro Precision, Macro Recall and Macro F1-Score are presented for each configuration (See Figure 6). This analysis allows for partitioning the performance improvement across the different methodological elements to determine the performance gains for each.

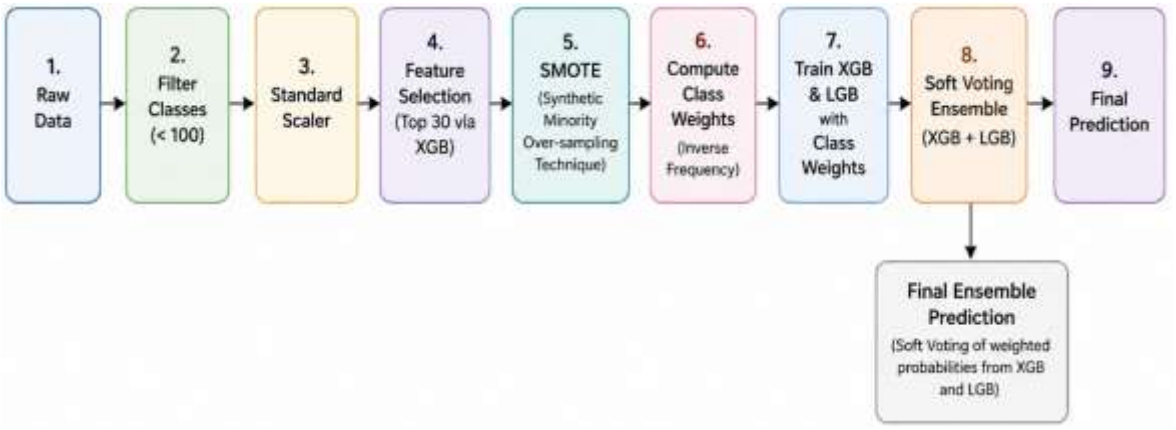

Figure 6: Detailed Architecture of Phase 2

2.6. Mitigation Simulation Framework

The methodology incorporates a simulation of a mitigation in order to narrow the gap between detection and the practical aspects of cyber-defense. This simulation simulates a real-world defensive response that is caused by the output of the ensemble model:

- Attack Traffic Modeling: A synthetic attack pattern is generated over a number of discrete time-steps with varying attack intensities (e.g., Poisson or Exponential distribution).
- Confidence-Based Blocking: The system implements a blocking policy that evolves according to the confidence. When the ensemble model is confident about a particular traffic segment (e.g. > 0.65), a large percentage (e.g. 85%) of the malicious packets detected by the ensemble model are considered "blocked" by the mitigation module and the rest are considered "passed.When ensemble model has a high confidence score about a specific traffic segment (e.g. > 0.65), a large percentage (e.g. 85%) of the malicious packets detected by the ensemble model are marked as “blocked” by the mitigation module and the rest are marked as “passed”.
- Blocking Efficiency Metric: Blocking Efficiency is a measure of the overall effectiveness of the mitigation, which is given by the ratio of blocked packets to total number of attack packets generated over the time of the simulation. This one quantifies the readiness for deployment of the proposed system.

2.7. Implementation and Validation Setup

Implementation of the whole framework in Python with the following libraries: Scikit-learn, XGBoost, LightGBM, imbalanced-learn, NumPy, Pandas, and Matplotlib/Seaborn. After the reproducible 20% subset of the data are sampled and rare classes are filtered, the remaining data are split into 80% for training and 20% for testing, randomly seeded at 42. All preprocessing operations that depend on training are fitted only on the training partition. The training set is used to estimate the parameters for standardization and they are then applied to the held-out test set. Also, the feature selection algorithm is executed on the training partition. The test data are not oversampled and SMOTE works only with the training data.

The final model is tested on the test partition which is not used for training, and various evaluation metrics such as Accuracy, Precision, Recall, F1-Score, Macro F1-Score and per-class performance metrics are calculated. Macro F1-Score and per-class Recall are used as the main performance measures since the main research goal is the reliable recognition across imbalanced categories of attack.

III. Results and Discussion

This section provides a full assessment of the proposed two-phase detection and mitigation plan. The assessment is carried out on a clean held-out test set (20% of the CIC-IDS2017 sample data) to provide an unbiased generalization performance. Our analysis is divided into two parts; baseline performance (Phase 1) provides a baseline and is examined in detail, followed by a detailed examination of the proposed enhanced methodology (Phase 2), and a comparative discussion that shows the trade-offs and improvements that have been made.

3.1. Baseline Performance (Phase 1)

The baseline methodology used three individual classifiers with a full 15-class dataset using the adaptive SMOTE strategy.The baseline methodology was based on three individual classifiers: Random Forest (RF), XGBoost (XGB), and LightGBM (LGB), with an adaptive SMOTE strategy applied to the full 15-class dataset. Overall performance statistics for these models are provided in Table 3. Random Forest outscored the other two with the highest accuracy (99.83%), whereas XGBoost performed the best in a macro-level assessment with a Macro F1-Score of 0.7742. The high accuracy for all the models (0.99–0.999) contrasted with the relatively low Macro F1-Scores (0.58–0.77) suggest a high-class imbalance effect, as the dominant class (e.g., BENIGN, DDoS) dominates the average.

Table 3: Performance Comparison of Baseline Models (Phase 1)

Model	Accuracy (%)	Macro Precision	Macro Recall	Macro F1-Score
Random Forest	99.83	0.7354	0.7353	0.7353
XGBoost	99.64	0.7604	0.8175	0.7742
LightGBM	97.11	0.5257	0.7973	0.5821

The confusion matrix for the best baseline model (Random Forest) is shown below in Figure 7. The matrix shows a high degree of classification accuracy for the majority classes (BENIGN, DDoS and PortScan), but it uncovers a serious defect in the detection of extremely rare attack categories. In particular, classes like Heartbleed, Infiltration and Web Attack – Sql Injection feature supports of zero or one in the test set and get a F1-Score of zero (as explained in the per-class breakdown of the best model). The model's generalization to the patterns that are not well represented is demonstrated by the F1-Score of 0.508 even for moderately rare classes such as Web Attack – XSS.

Figure 7. Confusion Matrix of the Best Baseline Random Forest Model

The class distribution before and after applying adaptive SMOTE in Phase 1 is shown in Figure 8. It is clearly seen that SMOTE was able to balance the majority classes, but for the smallest class (with only 2 samples in the train split), it used a dynamic $k_{neighbors}$ =1 variance and do not form a good decision boundary. Hence the baselines models have a high variance across the classes, which means that the use of a simple application of SMOTE on the complete 15-class set is not enough to cover the large-scale imbalanced network intrusion detection application.

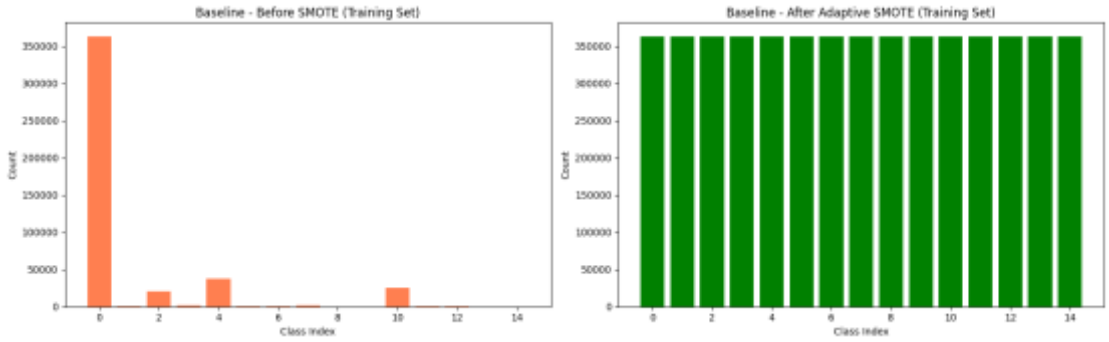

Figure 8. Class Distribution Before and After Adaptive SMOTE in Phase 1

3.2. Proposed Methodology Performance (Phase 2)

The proposed approach overcomes the drawbacks of Phase 1 by incorporating a rare-class filtering, feature selection and class weighting to create a soft-voting ensemble. The final ensemble performance on the smaller 12-class dataset is summarized in Table 4 and detailed performance breakdown per attack category.

The proposed model achieves an overall accuracy of 99.51% and a Macro F1-Score of 0.8509 which is significantly better than baseline in terms of the balanced classification. The near-perfect score of 0.9956 on the weighted F1-Score verifies that the model can still achieve a near-perfect detection rate for the high-frequency classes while also improving the rate of detection for the minority attack patterns.

Table 4: Per-Class Performance of the Proposed Model (Phase 2)

Class	Precision	Recall	F1-Score	Support
BENIGN	0.9998	0.9947	0.9972	90852
Bot	0.3737	1.0000	0.5441	71
DDoS	0.9977	0.9971	0.9974	5128
DoS GoldenEye	0.7527	0.9976	0.8580	415
DoS Hulk	0.9891	0.9986	0.9938	9297
DoS Slowhttptest	0.9862	1.0000	0.9930	214
DoS slowloris	0.9357	0.9915	0.9628	235
FTP-Patator	1.0000	0.9938	0.9969	323
PortScan	0.9925	0.9991	0.9958	6368
SSH-Patator	0.9915	0.9957	0.9936	235
Web Attack – Brute Force	0.4324	0.5333	0.4776	60
Web Attack – XSS	0.2750	0.7333	0.4000	30
Macro Avg	0.8108	0.9409	0.8509	113228

The proposed framework significantly enhances the general Macro F1-Score, but there is a certain heterogeneity among the traffic categories in the retained traffic. Specifically, Web Attack–Brute Force and Web Attack–XSS achieve F1-Scores of 0.4776 and 0.4000, respectively, indicating that it is difficult to differentiate between attack patterns that are not very well represented even after the minimum-support filtering. The Bot class shows a different behavior, with a high recall (1.0000) but a relatively low of precision (0.3737). This means that the model focuses on the identification of potential instances of Bot and thus has a relatively high rate of false positive assignments. A recall behavior may be desirable in intrusion detection, where not detecting malicious traffic could have more of a security impact than additional alerts. However, the results of this experiment show that the class imbalance is not fully removed by SMOTE and class weighting methods, and different techniques to address the problem may be needed in order to achieve better results, possibly by using more complex attack representations or further tests on independently captured network traffic.

The confusion matrix of the ensemble model is shown in figure 9. Off diagonal misclassifications are significantly improved from Phase 1. Of particular note, the model's performance for the Bot class is perfect recall (1.00) with lower precision (0.37), meaning that the threshold it uses for its classification is conservative in favor of recalling every Bot

instance, which also means that there are a few false positives. This compromise might sometimes be acceptable in cases of security attacks, where it is better to miss a real attack than to trigger a false alarm.

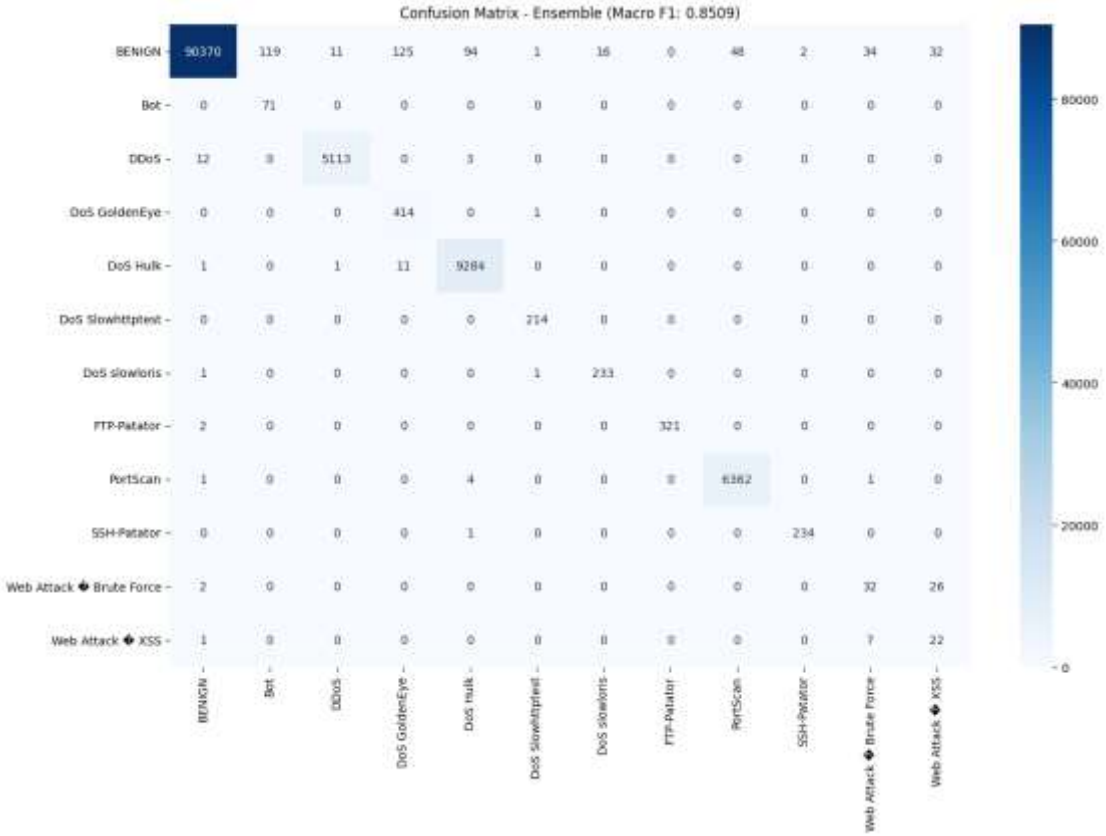

Figure 9. Confusion Matrix of the Proposed Ensemble Model in Phase 2

The feature selection mechanism effectively decreased the number of features from 77 to top 30 most informative features. Figure 10 shows the feature importance scores, which indicates that the network flow duration, packet length statistics, and certain TCP/UDP flags are the most important features, while others which are noisy and redundant have been successfully pruned. This reduction will speed up the inference process, and it will also serve as a regularize, which can help to improve the generalization of the ensemble on the test sett.

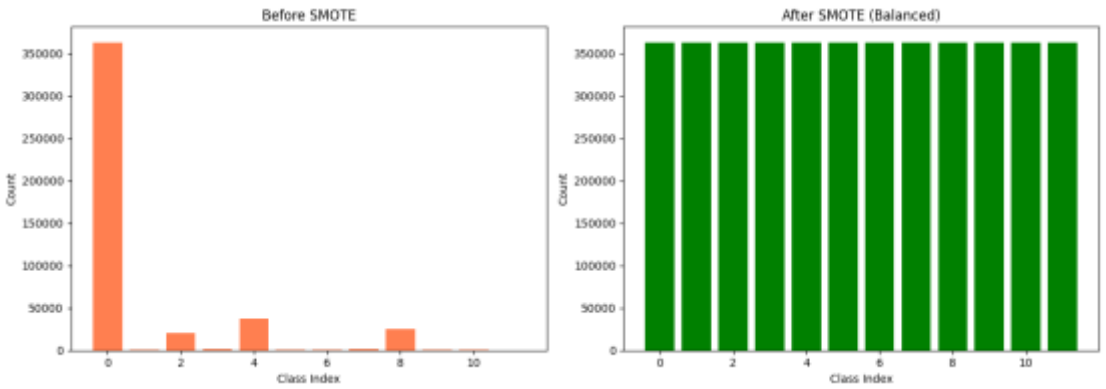

Figure 10. XGBoost Feature Importance of the Top 30 Selected Features

Lastly, Figure 11 simulates the attack mitigation process triggered by the ensemble's confidence. A simulated blocking efficiency of 72.23% is obtained for the model when a block occurs when the detection is high confidence. As can be seen in the mitigation curve, attack traffic drops off dramatically following the activation point, indicating that the detection system can be successfully plugged into a real-time defensive response system (See Figures 11 & 12).

The mitigation experiment should be seen as a proof-of-concept experiment to evaluate the integration between detection confidence and a defensive reply, not as a real-world packet-blocking test. This simulated blocking efficiency is the result of the attack strategy defined in Section 3.5, the synthetic attack, the confidence threshold, and the blocking assumptions of the proposed confidence-based response. Thus, this result shows the feasibility of linking the classifier to an automatic mitigation logic, but no validated mitigation rate has been achieved in production network conditions. Real-world validation would involve operation (or replay) of representative traffic on an operational network control mechanism and the measurement of other measures, including false blocking, time to respond, throughput effects, and recovery time from an attack.

Figure 11. Simulated Attack Mitigation Performance Based on Ensemble Confidence

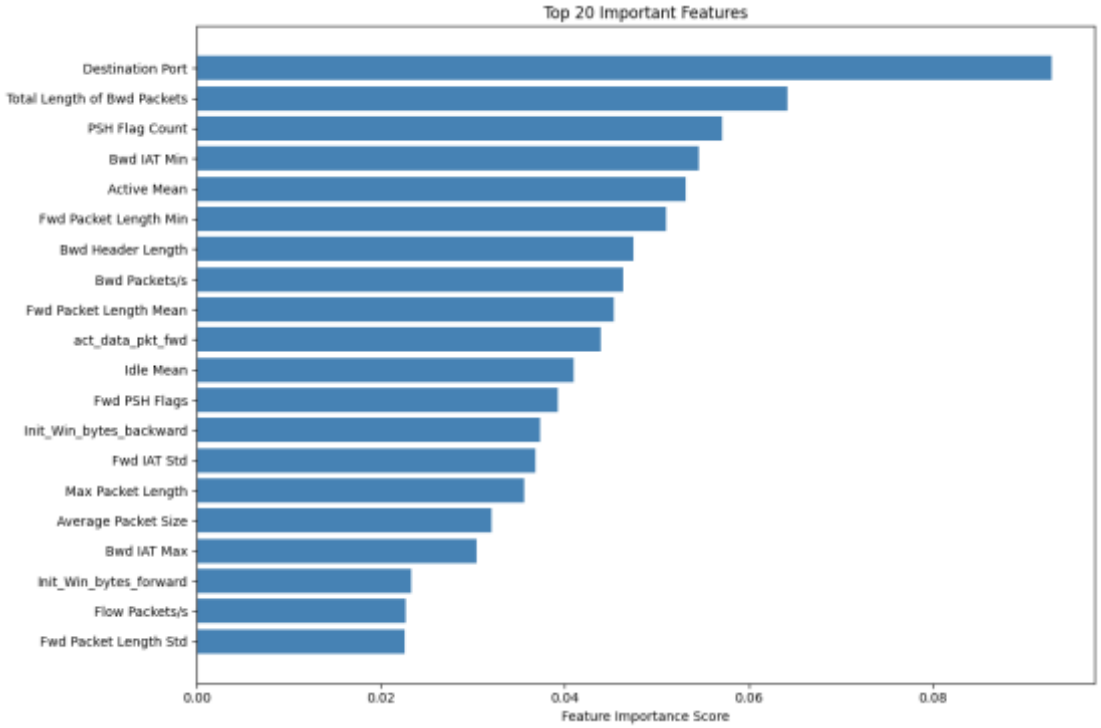

Figure 10. Top 20 Most Important Features Selected by XGBoost

3.3. Comparative Analysis and Discussion

Table 5 shows a side-by-side comparison of the best outcome from Phase 1 and the proposed outcome from Phase 2. The most significant finding is that there is a trade-off between normal accuracy and balanced macro-level performance. The baseline Random Forest had a slightly higher accuracy (99.83% vs. 99.51%) but with a considerably smaller Macro F1-Score (approx. 0.32 percentage points). The proposed methodology brings the Macro F1 from 0.7353 to 0.8509 (+11.56 percentage points absolute).

The slight drop in overall accuracy from 99.83% in the best Phase 1 model to 99.51% in the proposed Phase 2 framework does not necessarily indicate any loss in the detection capability. Since the imbalanced dataset is highly sensitive to the dominant traffic classes, the accuracy is significantly affected by these while Macro F1 treats all classes equally. The proposed framework not only improves the overall Macro F1 performance from 0.7353 to 0.8509 (absolute improvement of 0.1156) but also improves the Macro Recall from 0.7353 to 0.9409. This means that the proposed framework gives a more balanced recognition of the retained attack categories at the expense of a very slight drop in overall accuracy. Thus, the proposed framework is better evaluated in the current imbalanced classification scenario, rather than using accuracy as the sole metric for evaluation.

Table 5: Comparative Summary Between Baseline and Proposed Methodology

Metric	Phase 1 (Best Baseline: RF)	Phase 2 (Proposed Ensemble)
Accuracy	99.83%	99.51%

Macro Precision	0.7354	0.8108
Macro Recall	0.7353	0.9409
Macro F1-Score	0.7353	0.8509
Feature Space	77 features	Top 30 features
Classes	15	12 (Rare classes filtered)
Decision Strategy	Individual (RF)	Soft Voting (XGB + LGB)

The Phase 2 improvement is due to the use of rare-class filtering, training-only feature selection, SMOTE, class weighting, and soft-voting ensemble learning. The framework narrowed down the feature space to 30 informative variables and reached 99.51% accuracy, 0.8509 Macro F1, 0.9409 Macro Recall, and 0.9956 Weighted F1 (Macro F1 increased from 0.7353 in the baseline. But some minor classes, such as Web Attack–Brute Force, Web Attack–XSS, remained with low F1-Scores. The Bot class classifies all instances correctly, although there are some misclassifications. The simulated mitigation was 72.23% blocking efficiency, which should be considered a proof of concept for the actual blocking efficiency, not in the real world. The performance improves over the recent DDoS detection and mitigation studies, not just through accuracy but by taking into consideration the multiclass performance under severe class imbalance. Several limitations remain. Evaluation is based on CIC-IDS2017, and only generalization conclusions can be made for unseen networks. The 10% sampling could under-represent rare traffic patterns, and the classes that had fewer than 100 observations were not included in Phase 2 as they were not sufficiently supported to perform a reliable evaluation. Furthermore, the mitigation component is simulated and relatively low F1-Scores are seen in some minority classes.

The framework will be tested with independent datasets and cross-dataset scenarios, alternative imbalance strategies will be explored, threshold optimization and probability calibration will be studied, and repeated experiments with random seeds will be conducted in the future. The mitigation component will also be expanded to traffic replay and real network level testing to evaluate blocking accuracy, false positives, response latency and computation overhead.

Conclusion

This study designed and tested an integrated machine learning framework to detect a multiclass DDoS attack and simulate-based mitigation by leveraging the CIC-IDS2017 benchmark dataset. The methodology was conceived to tackle important limitations in network intrusion detection such as strong class imbalance, high dimensional feature spaces, redundant information, and the use of accuracy as a measure of performance alone. A two-phase experimental design was used to allow a controlled comparison of the conventional baseline and an enhanced methodology. The baseline methods used were Random Forest, XGBoost and LightGBM with adaptive SMOTE, while the approach proposed in this paper combined rare-class filtering, training-only feature selection, SMOTE, class weighting, and soft-voting ensemble learning. The experimental results indicate that the proposed design is effective in enhancing balanced multi-class recognition. The overall accuracy of the baseline Random Forest was found to be 99.83%, but its Macro F1-Score was 0.7353. The proposed ensemble was found to be 99.51% accurate, 0.8108 Macro Precision, 0.9409 Macro Recall, 0.8509 Macro F1-Score and 0.9956 Weighted F1-Score. In the light of this, the percentage of Macro F1 improved by 11.56 and that of Macro Recall rose from 0.7353 to 0.9409. The

results show that the proposed framework provides significantly better-balanced recognition performance across the retained attack categories, although there is a slight drop in recognition accuracy. The feature-selection phase also whittled the feature space down from 77 to 30 informative variables, promoting computational efficiency and information redundancy reduction. In addition, the mitigation simulation had a 72.23% blocking efficiency, which showed the possibility of associating detection confidence with an automatic response to the protection. But this finding is a proof-of-concept simulation not validated real-world mitigation performance. In general, the results suggest that the use of imbalance-aware preprocessing and ensemble learning can offer a more robust foundation for robust DDoS detection. The framework should be validated in the future with independent data, cross-dataset experiments, bigger traffic sets and actual or replayed mitigation scenarios.

References

1. Yungaicela-Naula, N. M., Vargas-Rosales, C., Pérez-Díaz, J. A., Jacob, E., & Martinez-Cagnazzo, C. (2023). Physical assessment of an SDN-based security framework for DDoS attack mitigation: Introducing the SDN-SlowRate-DDoS dataset. *IEEE Access, 11*, 46820–46831.
2. Alashhab, A. A., Zahid, M. S., Isyaku, B., Elnour, A. A., Nagmeldin, W., Abdelmaboud, A., ... & Maiwada, U. D. (2024). Enhancing DDoS attack detection and mitigation in SDN using an ensemble online machine learning model. *IEEE Access, 12*, 51630–51649.
3. Chouikik, M., Ouaissa, M., Ouaissa, M., Boulouard, Z., & Kissi, M. (2024). Detection and mitigation of DDoS attacks in SDN based intrusion detection system. *Bulletin of Electrical Engineering and Informatics, 13*(4), 2750–2757.
4. Wang, K., Fu, Y., Duan, X., & Liu, T. (2024). Detection and mitigation of DDoS attacks based on multi-dimensional characteristics in SDN. *Scientific Reports, 14*(1), 16421.
5. Alemayehu, M., Ghanem, M. C., Kheddar, H., Dunsin, D., & Lacerda, M. J. (2026). Systematic analysis on the use of AI techniques in industrial IoT DDoS attack detection, mitigation, and prevention. *IoT, 7*(3), 51.
6. Odeh, A., Aboshgifa, A., & Belhaj, N. (2023, November). Mitigating DDoS attacks in cloud computing environments: Challenges and strategies. In *2023 International Conference on Electrical, Computer and Energy Technologies (ICECET)* (pp. 1–4). IEEE.
7. Rios, V. D. M., Inácio, P. R. M., Magoni, D., & Freire, M. M. (2022). Detection and mitigation of low-rate denial-of-service attacks: A survey. *IEEE Access, 10*, 76648–76668.
8. Alsumaidaie, M. S. I., Alheeti, K. M. A., & Alaloosy, A. K. (2024). An assessment of ensemble voting approaches, random forest, and decision tree techniques in detecting distributed denial of service (DDoS) attacks. Iraqi Journal for Electrical and Electronic Engineering, 20(1), 16–24.
9. Djuitcheu, H., Shah, T., Tammen, M., & Schotten, H. D. (2023, November). DDoS impact assessment on 5G system performance. In 2023 IEEE Future Networks World Forum (FNWF) (pp. 1–6). IEEE.
10. Fotse, Y. S. N., Tchendji, V. K., & Velempini, M. (2024). Federated learning based DDoS attacks detection in large scale software-defined network. IEEE Transactions on Computers, 74(1), 101–115.

11. Hassan, A. I., El Reheem, E. A., & Guirguis, S. K. (2024). An entropy and machine learning based approach for DDoS attacks detection in software defined networks. Scientific Reports, 14(1), 18159.
12. Ibrahim, A. J., Répás, S. R., & Bektaş, N. (2025). Feature-optimized machine learning approaches for enhanced DDoS attack detection and mitigation. Computers, 14(11), 472.
13. Saiyed, M. F., & Al-Anbagi, I. (2024). A genetic algorithm-and t-test-based system for DDoS attack detection in IoT networks. IEEE Access, 12, 25623–25641.
14. Sambangi, S., Gondi, L., & Aljawarneh, S. (2022). A feature similarity machine learning model for DDoS attack detection in modern network environments for Industry 4.0. Computers & Electrical Engineering, 100, 107955.
15. Shanthi, A. L., Kumar, R., Hemalatha, B. M., Mallick, S., Srinivas, K. G., & Mahalakshmi, G. (2026). An intelligent fuzzy graph-based framework for network threat detection, cyber risk assessment, and adaptive mitigation. International Journal of Computer Information Systems and Industrial Management Applications, 18(17s), 1098–1123.
16. Singh, C., & Jain, A. K. (2024). A comprehensive survey on DDoS attacks detection & mitigation in SDN-IoT network. e-Prime—Advances in Electrical Engineering, Electronics and Energy, 8, 100543.
17. Tiwana, P. S., Kaur, L., Kaur, N., Wadhwa, H., Kumari, N., & Sharma, R. (2024). Mitigating DDoS attacks: A distributed blockchain-SDN secure IoT system enhanced by artificial neural networks. Computational Methods in Science and Technology, 30(3), 467–472.
18. Alrumaih, T. N., & Alenazi, M. J. (2025). ERINDA: A novel framework for enhancing the resilience of industrial networks against DDoS attacks with adaptive recovery. Alexandria Engineering Journal, 121, 248–262.
19. Alshdadi, A. A., Almazroi, A. A., Ayub, N., Lytras, M. D., Alsolami, E., & Alsubaei, F. S. (2024). Big data-driven deep learning ensembler for DDoS attack detection. Future Internet, 16(12), 458.
20. Danang, D., & Mustofa, Z. (2026). CLSTMNet architecture: A CNN–LSTM-based hybrid deep learning model for DDoS attack detection and mitigation in network security. Journal of Artificial Intelligence and Technology, 6, 207–214.
21. Joo, S., Park, S. H., Shim, H. Y., Oh, Y. S., & Lee, I. G. (2025). Machine learning-based detection and selective mitigation of denial-of-service attacks in wireless sensor networks. Computers, Materials & Continua, 82(2), 2475–2494.
22. Khalil, K., Hansen, L. B., Tayebi, E., & Gehrmann, C. (2025, November). Learning at the edge: Simulated DDoS detection in 5G networks. In 2025 38th Conference of Open Innovations Association (FRUCT) (pp. 136–143). IEEE.
23. Koksal, S., Catak, F. O., & Dalveren, Y. (2024). Flexible and lightweight mitigation framework for distributed denial-of-service attacks in container-based edge networks using Kubernetes. IEEE Access, 12, 172980–172991.
24. Olufemi, D., Ejiade, A. O., Ikwuogu, F. O., Olufemi, P. E., & Bobie-Ansah, D. (2025). Securing software-defined networks (SDN) against emerging cyber threats in 5G and future networks: A comprehensive review. International Journal of Engineering Research.
25. Qin, X., Doss, R., Jiang, F., Qin, X., & Long, B. (2025). Securing ICS networks: SDN-based automated traffic control and MTD defensive framework against DDoS attacks. Computer Communications, 241, 108252.

26. Singh, A., Kaur, N., & Kaur, H. (2022). An extensive vulnerability assessment and countermeasures in Open Network Operating System software-defined networking controller. Concurrency and Computation: Practice and Experience, 34(15), e6978.
27. Varalakshmi, I., & Thenmozhi, M. (2025). Entropy-based earlier detection and mitigation of DDoS attack using stochastic method in SDN-IoT. Measurement: Sensors, 39, 101873.
28. Verma, A., Saha, R., Kumar, G., Conti, M., & Kim, T. H. (2024). PREVIR: Fortifying vehicular networks against denial of service attacks. IEEE Access, 12, 48301–48320.
29. Clinton, U. B., Hoque, N., & Robindro Singh, K. (2024). Classification of DDoS attack traffic on SDN network environment using deep learning. Cybersecurity, 7, 23.