International Journal of Engineering Trends and Technology
ISSN: 2231-5381/ https://doi.org/10.14445/22315381/IJETT-V74I7P115

Volume 74 Issue 7, 217-229, July 2026

Original Article

Analyzing Class-Level Performance Variability in Multi-Class DDoS Detection: A Diagnostic Framework Based on XGBoost and Per-Attack Difficulty Scoring

Keano Nikko L. Sy[1], Patrick D. Cerna[2]

[1,2]*Graduate School, State University of Northern Negros, Sagay City, Negros Occidental, Philippines.*
[1]*FCSIT, Southern Leyte State University - Main Campus, Sogod, Southern Leyte, Philippines.*

[1]*Corresponding Author : ksy@southernleytestateu.edu.ph*

Received: 06 January 2026 Revised: 10 February 2026 Accepted: 18 June 2026 Published: 28 July 2026

Abstract - *The DDoS attacks continue to be one of the most widespread menace to the modern network infrastructure, since they flood systems with harmful traffic and disabling of the valid services. Considering the continuously changing nature of the attack pattern, detection through a machine-learning method has been a mandatory requirement to observe the slight differences that are present between the different kinds of DDoS attacks. The paper trains and tests a multi-class XGBoost detection model on a mixed dataset with eleven different categories of DDoS attacks. Precision, Recall, F1 -score, ROC -like interpretations, Precision -Recall curves, confusion matrices, and computational efficiency metrics were used to evaluate the model. The results indicate that there is a high level of detectability variation among classes. High-performance attacks, including DrDoS NTP, TFTP, and DrDoS MSSQL, had almost perfect F1 -scores, which means that they were very separable and displayed consistent patterns of features. NetBIOS, SNMP, and Syn were moderate in their performance, with some overlap in the distributions of features. Classes with the worst performance of LDAP, SSDP, DrDoS UDP, DrDoS DNS, and UDPLag had more misclassification rates and difficulty scores, indicating complex or noisy traffic characteristics that do not help with accurate classification. The model was also found to be very computationally efficient, and inference times were fast enough to allow the model to be used in a real or near-real-time setting. These findings highlight the importance of using hybrid datasets and diagnostics of class-level performance to reveal variability of attack detection. The study concludes that XGBoost has strong flexibility in the accuracy, stability, and working efficiency in multi-class DDoS detection. The next improvements can include the addition of deep-learning frameworks, adversarial training, and real-time threat feeds, which can be used to enhance detection and other network defense against the most difficult types of attacks.*

Keywords - *XGBoost Classification, Multi-Class Intrusion Analysis, Network Security, Machine Learning.*

1. Introduction

Distributed Denial-of-Service (DDoS) attacks remain among the most prevalent cybersecurity threats to the modern digital infrastructure. These attacks flood servers, networks, and cloud-based services with excessive malicious traffic, thus causing service unavailability, loss of money, and disruption of operations. Recent publications note that the size and quality of DDoS attacks are continuously growing due to the supply of Internet of Things (IoT) devices, the growth of large-scale botnets, and the presence of high-speed network links [1, 2]. As evidenced by the contemporary intrusion-detection datasets, such as CIC DDoS 2019, UNSW-NB15, and Bot-IoT, the present-day DDoS attacks use different strategies to simulate legitimate traffic behavior, including reflection-based amplification, exploitation of protocols, and low-rate flooding [3-5]. This has, in turn, rendered the old signature-based and threshold detection systems very inefficient in the face of the malicious traffic, which is very similar to that of the normal network traffic [6, 7].

In order to address these limitations, intrusion detection systems that rely on machine-learning technologies have become widely used, as they have the ability to learn the complex, non-linear relationships between high-dimensional network-traffic data. Ensemble-based techniques, such as Random Forest, Gradient Boosting, LightGBM, and XGBoost, have shown good detection rates through the utilization of multi decision boundaries to represent a variety of attack features [8, 9]. Simultaneously, it has been suggested that deep-learning methods with recurrent neuron networks, convolutional, and attention schemes can be employed to model both time and space dependencies of the traffic patterns [10, 12]. More recent work has investigated more advanced paradigms like federated learning, adversarial robustness, and

multi-agent detection models to improve scalability and resilience in distributed settings [13, 14].

Research has shown that in recent times, the field of multi-class DDoS detection has achieved a lot by using ensemble learning and deep-learning frameworks. Random Forest, Gradient Boosting, LightGBM, and XGBoost ensemble-based methods have demonstrated strong capability to identify non-linear relationships among network-traffic features and thus better detection of various types of attacks [8, 9]. At the same time, deep-learning-based approaches, such as recurrent neural networks, convolutional neural networks, and hybrid solutions, have also emerged as the modeler of temporal relationships and more complex traffic patterns in a multi-class intrusion-detection context [10, 12, 19]. The methods are especially useful in conditions that can be described by large traffic flow and diverse attack patterns.

More recent works published in 2024-2025 continue to enhance the capabilities of multi-class detection of DDoS as they add adaptive learning systems, federated designs, and adversarial resilience. Research on SDN networks and clouds highlights how multi-dimensional traffic properties and hybrid ensemble frameworks are useful to improve overall detection functionality [7, 22]. It has also been suggested to use advanced frameworks based on federated learning, multi-agent coordination, and blockchain-assisted defense to enhance the scalability, robustness, and privacy of distributed detection systems [13, 14]. The main objectives of these approaches would be to increase the accuracy of system-wide detection and resilience against the changing attack strategies.

However, despite modern research showing an improved global performance, most of the studies evaluate the detection efficacy using aggregate measures, such as overall accuracy, macro-average F1 20, or ROC-AUC [8, 16, 20]. Some studies also provide per-class accuracy and recall values, but these are rarely presented in a systematic way, and do not attempt to analyze misclassification behavior or interpret the occurrence of consistent underperformance in a particular category of attack [15, 17]. UDP-based, low-rate, and protocol-mimicking DDoS attacks, specifically, still utilize features that overlap and similarity with regular traffic; however, the problem has not been studied extensively [18, 23]. On this basis, the available literature offers little diagnostic evidence on class-specific detection difficulty and offers few instructions on how to improve the models.

In spite of these developments, one issue with the multi-class detection of DDoS-type attacks that persists is the inability to detect such attacks equally. Empirical evidence has always noted that some types of DDoS attacks achieve almost a perfect detection, and others have significantly lower accuracy, and Recall, even using the same models and data sets [15, 16]. That imbalance is exacerbated by an imbalance in classes, overlapping features, and the fact that specific instances of attacks and authentic traffic are close to each other [17]. Specifically, low-rate, UDP-based, and protocol-mimicking attacks are more likely to generate sparse or noisy traffic patterns that cannot be discriminated against other classes and therefore raise the misclassification rates [18].

The main flaw of the existing literature is that the model is assessed using aggregate performance measurements, such as overall accuracy, macro-averaged F1-score, or ROC-AUC [8, 16]. Even though these measures provide an overview of global detection ability, they mask large amounts of variation at the level of classes. Aggregation can be misleading in operational intrusion-detection systems: a model might seem very effective in general and at the same time be unable to identify a subset of attack types that represent a serious operational threat. Although some studies provide per-class statistics, they rarely perform systematic investigations to explain the phenomenon of misclassification or feature similarity, as well as similar determinants of detection difficulty by attack type [15, 17].

Recent literature, which uses deep-learning-based methods and ensemble-based methods, has attained improved net-detention rates by combining time-based modelling, blended models, or active learning methods [10, 14]. However, the main aim of these studies is to improve performance worldwide and cannot provide much diagnostic value to understand why specific classes of attacks are so hard to detect. Although there is a deficit of formal mechanisms to quantify and compare the relative difficulty of detecting individual types of DDoS attacks, even when studies are foreground explainable or per-class reported. To this end, there is little information provided in the literature that can guide how to customize feature engineering, dataset construction, or optimization of models to overcome class-specific drawbacks in the multi-class intrusion-detection systems.

To address this, the given paper presents a diagnostics framework based on the class level to detect multi-class DDoS using an XGBoost-based classes detector. Instead of presenting a new detection algorithm, the value of this paper lies in its analysis, which tries to uncover and describe the variability of performance among different types of DDoS attacks. The framework involves the Per-Attack Difficulty Score (ADS), a quantitative score that was created to indicate the relative detection difficulty of a single attack class, based on the performance results of the classes. The framework produces a more detailed understanding about the aspects that make some attacks (e.g., LDAP, SSDP, and UDP-based floods) more evasive all the time by supplementing traditional measures of evaluation with ADS, misclassification analysis, and precision-recall dynamics. By moving the assessment of the aggregate accuracy to per-class diagnostic evaluation, the paper will provide a practical and interpretable approach to the evaluation of multi-class DDoS detection systems. The

findings thus obtained give practical information on the development of improved feature selection, data balancing, and model optimization that would enhance the creation of more robust and threat-sensitive intrusion-detection systems across cloud scenarios, enterprise networks, software-defined environments, and IoT ecosystems. These attacks are more difficult than the high-volume amplification attacks.

This work provides an interpretable and practical framework in the evaluation of multi-class DDoS detection systems by shifting the evaluation focus on aggregate accuracy to per-class diagnostic evaluation. The findings provide practical recommendations on how to improve feature selection, data balancing, and model optimization to progress the creation of more resilient and threat-conscious intrusion-detection systems that would be applicable to cloud environments as well as enterprise networks, software-defined networks, and IoT ecosystems.

1.1. Objectives of the Study

The main objective of the research is to examine the performance variation of a multi-class DDoS detection model through behavioral dynamics of different types of attacks under machine-learning classification. The research assesses the advantages and weaknesses of an XGBoost-based model and attempts to explain why the different forms of attack have different detection rates. In that respect, a unified dataset of varied DDoS attacks is compiled through a systematic pipeline encompassing cleaning, normalization, and sampling, and filtering of features to guarantee accuracy and consistency. Model evaluation uses standard measures of performance, namely, accuracy, precision, Recall, F1 score, and confusion patterns, in order to ensure that training was successful and that performance was acceptable.

Based on these findings, the paper presents a Per-Attack Difficulty Score, a diagnostic scheme that attempts to predict the relative difficulty of each type of attack, based on the patterns of misclassifications, class imbalance, and similarity of features. The broad objective is to better understand the class-level behavior when DDoS are being detected and give a real-world contribution to improve the feature selection, data balancing, and model optimization. The end result of this contribution is the development of more reliable, threat-aware intrusion-detection systems in cloud-based environments, enterprise networks, software-defined systems, and Internet of Things systems.

2. Materials and Methods

2.1. Research Design

The present study will take an experimental quantitative research design to investigate the performance attributes of a multi-class DDoS detection model. The design enables stringent and thorough testing of machine-learning methods on carefully curated and standardized data, thus permitting objective testing of model behavior in a particular set of attack types. An XGBoost-based classifier was trained and tested by a controlled experiment, where accuracy, precision, Recall, F1-score, and confusion-matrix trends were used in the evaluation of the strengths and weaknesses of the model in identifying different DDoS attacks. The methodology enables the investigation of the variability of the classes, which is the focus of the research, and the creation of the difficulty-scoring system that clarifies the comparative difficulty of recognizing each type of attack. In contrast to descriptive or qualitative designs, this method assumes the presence of empirical data and numerical analysis, which is a valid basis of assessing the performance of classification and explaining the diagnostic value of per-attack analyses.

2.2. Data Sources and Sampling

The paper takes advantage of publicly available datasets of actual network-traffic records that represent a variety of DDoS attacks. All the major sources are some of the most famous intrusion-detection datasets, CIC DDoS 2019, CIC DoS 2017, CIC IDS 2018, UNSW NB15, and Bot IoT, where the labeled traffic flows are offered that relate to reflection attacks, flooding attacks, and low-rate attacks. Every dataset consists of numerical characteristics embedded in raw network flows, which allows machine-learning models to evaluate the differences in behaviors among the types of attacks. A network-flow record is the unit of analysis, which is not a human subject. In order to cope with the huge amount of traffic data, a block-based loading mechanism was used: files were loaded in blocks, filtered and sampled in order to drop corrupted values and maintain the numeric integrity. A dataset of several hundred thousand valid records was formed after the preprocessing, and the following types of attacks were represented: DrDoS DNS, DrDoS NTP, DrDoS MSSQL, DrDoS LDAP, DrDoS UDP, SSDP floods, SYN floods, TFTP floods, and UDP Lag attacks. The under-represented types of attacks were randomly sampled in each category, which helped avoid class imbalance and minimize sampling bias and made sure that the model could learn the specificity of traffic related to each type of attack in adherence to multi-class intrusion-detection principles.

2.3. Feature Selection and Preprocessing

The feature set used in this study consists of numerical flow level traffic statistics derived from network packet captures, including duration-based metrics, packet counts, byte counts and statistical descriptors of inter-arrival times and packet sizes. These features were selected because they directly characterize traffic behavior patterns that differ across DDoS attack types, such as amplification intensity, reflection behavior and protocol-specific flooding dynamics. For example, amplification-based attacks typically generate high packet and byte rates with low variability, while low-rate or protocol abuse attacks exhibit more irregular temporal patterns. Non-behavioral and non-generalizable attributes such as flow identifiers, timestamps, IP addresses, port

numbers and protocol labels were removed prior to training. These fields encode dataset-specific or environment-dependent information that does not contribute to learning attack behavior and may lead to overfitting. Only numeric features were retained to ensure consistency across learning algorithms and to avoid implicit encoding of attack labels.

Redundancy among features was controlled implicitly through the use of tree-based learning and regularization, which reduces reliance on highly correlated attributes. This strategy allows the model to prioritize the most discriminative features while limiting the influence of redundant or collinear measurements without the need for aggressive manual feature elimination.

2.4. Data Preprocessing and Instrument Validation

The processed multi-class DDoS data, obtained as a result of a systematic data-preparation process, was the main research tool in this research. Raw traffic data of various publicly available intrusion-detection repositories contained numerical flow features that needed to be refined thoroughly before being modelled. The preprocessing stage involved removing records with corruption or with missing or infinite values, normalization of the numeric fields, and harmonization of the column names across all the datasets to provide structural consistency. Further processing of removing duplicates, outlier clipping, and non-numerical attributes was done to maintain the integrity of the feature space. The study followed the research procedures of outlined intrusion-detection techniques to ensure the validity of the refined data, keeping only the network-flow features found in the previous literature as important clues of DDoS behaviour [16, 17]. Data whose labels were confused with attacks were eliminated to prevent incorrect classification in learning the model. The resulting output table consisted of a clean, consistent, and validated list of attack classes, which can be used to train the XGBoost-based multi-class detection model and can be used to perform a more in-depth performance analysis of each of the individual attack category.

2.5. Model Training Hyperparameters

In the case of the XGBoost classifier, a list of common hyperparameters was used to trade off expressiveness of the model and generalization. The tree maximum depth was set to eight, the learning rate was 0.1, and subsampling ratios, as well as column subsampling ratios, were both equal to 0.8. The use of histogram-based tree construction was to enhance the computational efficiency. The fixed number of boosting rounds was 200, and the training was monitored with the help of multi-class log loss in order to guarantee convergence.

Instead of performing a grid search, hyperparameters were initialized with suggested default values published in previous intrusion-detection research and adjusted by performing a few preliminary experiments. This method has been chosen so as not to over-optimize the overall accuracy measures and to ensure a steady analytical baseline in future per-class performance measurements and difficulty measures.

Class imbalance was reduced by training samples equally, and the weighted sample was used, making sure that the minority attack classes contributed equally to the optimization goal. This strategy, coupled with the natural regularization processes of the XGBoost model, minimized over-fitting and facilitated strong generalization with a wide range of types of DDoS attacks.

2.6. Per Attack Difficulty Score

The common metrics of evaluation, like overall accuracy, macro-averaged F1 score and weighted F1 score, are a global impression of the classifier performance, which tends to hide systematic deficiencies of the classifier in the form of individual attack categories. Even when overall performance is strong, in the case of multi-class DDoS detection, some types of attack might always be falsely classified either because of feature overlap, protocol similarity or class imbalance. In order to measure this phenomenon explicitly, an analytical diagnostic measure, Per Attack Difficulty Score, is proposed.

The Per-Attack Difficulty Score (ADS) for an attack class c is formally defined as follows:

$$ADS_c = 1 - F1_c$$

Equation (1) Per Attack Difficulty Score for class c

The fact that Equation (1) modeled the difficulty of the attack by modelling it as the complement of the effectiveness of classification shows that. and $F1_c$ is the class-specific F1 score that is received in class c, the F1 score is calculated as the harmonic mean of precision and Recall of that particular class. Precision denotes the ratio of correctly categorized samples in the sample of all samples with a prediction of class c, and Recall is the ratio of correctly categorized samples to all instances with a true value of class c.

As such, the formulation determines attack difficulty as the opposite of classification effectiveness. Because the F1 score simultaneously quantifies both false-positive and false-negative errors, the concept of difficulty of the F1c should be defined as being equal to 1 - $F1_c$, which would directly reflect the extent of misclassification to which an instance of an attack class belongs. The resulting ADS values are constrained between 0 and 1, with low values depicting more easily detectable attacks and high values depicting more difficult classification.

Per-Attack Difficulty Score (ADS) is perceived as a diagnostic measure and not as a performance goal, and is not

used in model training or optimization. Rather, ADS is calculated based on the performance of the different classes, in order to compare the types of DDoS attacks. As a way of quantifying the difficulty of each type of attack, ADS allows attack types to be directly ordered in terms of their detection difficulty, which is especially useful in the case of a highly imbalanced multi-class environment where a dominant attack type can always mask mis-classification of a minority or overlapping behavior class.

2.7. Data Collection Procedure

The process of data collection included three major steps. First, the internal sources consisted of publicly available intrusion-detection repositories, which package labelled network-traffic logs of many types of attacks, and large-volume datasets, such as CIC DDoS 2019, CIC DoS 2017, CIC IDS 2018, UNSW NB15, and Bot IoT. Second, the raw data files were preprocessed to eliminate corrupted data, purify blank values, normalize numerical values, and standardize labelling by sources. Third, after data cleansing, the processed data were divided into training and testing datasets in a ratio of 80:20, allowing the machine-learning model to be trained on the majority of records and retained a separate set to test the performance. All steps were executed with automated Python scripts, which are reproducible and minimize human error and consistency between experimental trials.

2.8. Model Training

The training phase of the model was devoted to the construction of an XGBoost-based multi-class classifier capable of discriminating between multiple types of distributed Denial-Of-Service (DDoS) attacks. XGBoost was chosen due to its thorough documentation and demonstrated effectiveness in intrusion-detection research and its capability to identify complicated relationships in high-dimensional network flow data. It was initiated by the conversion of the cleaned data into a DMatrix structure, which allowed storing and computing efficiently. In each record, there were numerical characteristics of the traffic behaviour and flow statistics, except that the labels indicate the type of DDoS attack.

XGBoost is a gradient-boosting algorithm whereby decision trees are built in series. The new tree is trained to correct the errors of the previous ensemble at every iteration. The model is optimized based on an objective function which is made up of a training loss term and a regularization term. The regularization term reduces the complexity of the model to discourage overfitting by the training loss, which measures the difference between the model predictions and the actual labels. This twofold goal provides the combination of predictive power and model simplicity, and generates strong generalization with the types of attack.

$$Obj = \sum_{i=1}^{n} \; l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \; \Omega(f_k)$$

Equation (2) General objective function of XGBoost for multi-class classification

where
$l(y_i, \hat{y}_i)$ is the loss function for record i,
y_i is the true class label,
$\hat{y}_i$ is the predicted output,
and $\Omega(f_k)$represents the regularization applied to each tree f_k.

Since there are numerous categories of DDoS attacks, the XGBoost uses the SoftMax function to transform the raw output scores into the probability of the class. This change allows assigning the correct category of the attack with the highest probability of occurrence to each record. The use of SoftMax makes it a complete probability distribution on the classes, ensuring a normalized probability distribution across all attack classes. In addition, the probabilistic framework allows an extra understanding of the patterns of misclassification; low confidence values may indicate overlapping feature patterns shared across multiple attack typologies. It also promotes the granular performance diagnostics, which allow evaluating the efficacy of the model in disaggregating individual attack classes.

$$p_k = \frac{e^{z_k}}{\sum_{j=1}^{C} e^{z_j}}$$

Equation (3) SoftMax function for multi-class output
Where
$p_k$is the predicted probability of class k,
$z_k$is the raw score of class k,
and $C$is the total number of classes.

Multi-class cross-entropy loss is also used during the optimization and is used to penalize errors in predictions by comparing the actual labels with the predicted probabilities. This loss objective is typical of multi-class classification since it relates directly to the quality of the predicted probability distribution. It not only forces the model to choose the correct class but also to place high probability in the correct class compared to the rest, which is desirable in the case of feature overlap or other such patterns between DDoS traffic classes.

$$l(y, \hat{y}) = -\sum_{c=1}^{C} \; y_c \log(\hat{y}_c)$$

Equation (4) Cross-entropy loss function

Where
$y_c$is the true class indicator and
$\hat{y}_c$is the predicted probability for class c.

XGBoost presents a regularization term that prevents the tendency of overfitting by restricting the complexity of trees and limiting the weight of leaves. This step decreases the

dependence of the model on noise or subtle differences in the training data- of special importance when dealing with. The regularization promotes the development of simpler tree models, which generalize well over a wide range of DDoS attack types, which guarantees the stability of performance even when the attack classes are of high variability or when some of the attack classes have small training samples.

$$\Omega(f_k) = \gamma T + \frac{1}{2}\lambda \sum_{j=1}^{T} w_j^2$$

Equation (5) Regularization term,
Where

T is the number of leaves,
w_j is the weight of leaf j,
γ controls the penalty for additional leaves,
and λ controls the penalty for large leaf weights.

When training, XGBoost calculates both the first and second derivatives of the loss at each example and then uses those derivatives to train additional trees at that point. Class weights were also used to mitigate the issue of class imbalance that would ensure that the minority classes like LDAP, DrDoS UDP and UDP-Lag are sufficiently represented in the learning process. The training was performed in several boosting rounds until convergence, as shown by a plateau in values of evaluation loss.

The XGBoost classifier was trained on a set of hyperparameters that was carefully selected to balance the model expressiveness and model generalization. The depth of the maximum tree was set to eight to represent non-linear interactions of features, and at the same time to restrict tree complexity.

This was achieved by setting the learning rate to 0.1 to ensure that the convergence remained constant with boosting iterations. The ratio of sub-sampling and column subsampling was both 0.8 to minimize the variance and to decrease over-fitting. Tree construction using histograms was used to improve efficiency in computations on large-scale flows.

Hyperparameter estimation was based on a default-plus-tuning approach as opposed to an exhaustive grid search. Initial values were selected as a recommendation of other related works in previous intrusion detection research, and later optimized by a small amount of preliminary testing to guarantee consistent convergence and equal per-class performance.

The final XGBoost model is a powerful multi-class predictor that is capable of effectively detecting diverse categories of DDoS attacks at a high-accuracy rate due to the joint of gradient-based learning, probability-based prediction, and strict regularization. Follow-up analysis of the variability of the performance of the classes and Per-Attack Difficulty Scores is based on this training methodology.

A stratified five-fold cross-validation strategy was used to guarantee a high level of experimental rigor and strength of the reported results in the process of model training and evaluation. The dataset was randomly divided into five mutually exclusive folds, maintaining the original class ratio in each of the folds. In every iteration, one was trained on four folds and the other fold was left to be used as validation, and then the metrics of performance were averaged over all folds.

To alleviate the effects of stochastic variation due to data sampling and shuffling and model initialization, each experiment was run five times, and the results of the experiment are reported as an average across the runs. To enable reproducibility, a fixed random seed was always used in the data partitioning, sampling and model initializing.

The cross-validation, feature selection, class-balanced weighting, and regularization that are built into XGBoost were used together to mitigate over-fitting. The performance of the model was assessed only on held-out validation folds and not on the training data to ensure that the reported metrics are indicative of the generalization ability and not memorization.

2.9. Ethical Considerations

The research did not involve any human subjects; therefore, there was no need to obtain formal ethical consent. However, the use of open-source data followed the principles of ethical research. All the data were obtained from publicly available repositories and were used solely for academic use. There was no sensitive personal information that was collected, processed or stored.

2.10. Reproducibility and Implementation Details

These experiments were written in Python 3.10. The main libraries used are NumPy-1.26, pandas 2.1, scikit-learn 1. 5, and XGBoost 2.0. Such packages have been chosen due to their stability and their widespread use in the studies of machine-learning-based intrusion detection. All the reported results were reported in a standard software environment to ensure uniformity.

Training and testing of models was carried out through a stratified five-fold cross-validation, using a fixed random seed, 42, to partition data, sample, as well as model initialization. All the experimental settings were repeated five times each, and all the performance metrics are given as arithmetic means of the repetitions. The information was separated at the flow level and, thus, no flow was present in both the training and validation partitions.

Experiments were executed on a workstation equipped with an Intel Core i5 processor, 20 GB of RAM, and a 64-bit Windows operating system. No GPU acceleration was used,

as all models were trained using CPU-based implementations. The entire preprocessing chain, the feature-selection protocol, model-training code, and evaluation code are accessible on demand, and will be made public with publication. Every single hyperparameter setting applied is clearly outlined in the methodology section.

3. Results and Discussion

The reported results are an average on five-fold cross-validation, three independent cross-validation runs/fold, to provide a sense of stability in the performed experiment and to reduce performance variability. Here, the section is used to describe the experiment on an XGBoost-based multi-class classifier on eleven types of Distributed Denial-of-Service (DDoS) attacks. The evaluation will include the common classification metrics, misclassification analysis, Precision Recall-Area Under the Curve (PR-AUC), efficiency and the proposed Per-Attack Difficulty Score (ADS). Tables and figures are organized in such a way that they provide a detailed diagnostic portrait of aggregate model performance and the detection behavior unique to classes.

In order to assess whether the reported results are stable, all experiments were repeated several independent runs with stratified five-fold cross-validation, with performance measures averaged to the runs. The standard deviation of the total accuracy and macro-averaged F1 score between these repeated runs was low, which indicates that the model behaves consistently and is not highly sensitive to random initialization and data splitting. This low variance means that the performance reported is statistically consistent, and it is not determined by a specific train-test split.

3.1. Baseline Model Comparison

In order to place the proposed XGBoost-based multi-class DDoS classifier in perspective, the results of the proposed algorithm were compared to two well-known baseline learning methods, Logistic Regression and Random Forest. These were chosen due to their common use in research in network intrusion detection, as well as because of their differing representational capabilities, extending all the way up to the linear decision boundaries, down to ensemble-based non-linear modeling. The comparison is done focusing on the general classification performance as opposed to the per-class performance optimization to provide a reasonable and understandable performance standard.

Table 1. Baseline model comparison using accuracy and Macro-Averaged F1 score

Model	Accuracy	Macro F1-score
Logistic Regression	0.4484	0.3049
Random Forest	0.8890	0.7674
XGBoost	0.8753	0.7692

As indicated by Table 1, the performance of Logistic Regression is significantly lower, hence emphasizing the weaknesses of the linear classifiers working on the modelling of complex and non-linear feature correlations, which are contained within the multi-class DDoS traffic. Conversely, the overall accuracy of Random Forest and its macro-averaged F1-score are high, highlighting the usefulness of ensemble-based decision-tree models in predicting heterogeneous attack behaviors in intrusion detection tasks. Though the Random Forest can achieve similar accuracy across the globe, XGBoost shows more consistent behavior at the class level and is easier to interpret to perform diagnostic analysis, which is the reason behind using this algorithm in the current study.

Also, XGBoost exhibits a greater level of cross-validation consistency between folds and repeated executions of the algorithm, which causes it to be better aligned to fine-grained per-class performance, as opposed to aggregate accuracy optimization.

XGBoost provides similar overall performance with an increased control of regularization and gradient-based optimization, which makes this algorithm especially suitable to fine-grained analytical analysis. Noteworthy, this study is not aimed at specifying the state-of-the-art detection accuracy, but a stable and expressive classifier appears as the premise to the detailed analysis of per-class performance. This decision makes it possible to investigate vulnerabilities peculiar to classes afterwards and synthesize the offered Per-Attack Difficulty Score (ADS), which is the main analytical innovation of the given work.

3.2. Per Class Classification Performance

Table 2 presents the precision, recall, F1 score, and support for each of the eleven DDoS attack classes. These metrics summarize the per-class performance of the XGBoost-based multi-class classifier.

Table 2. Performance of XGBoost across eleven DDoS attack classes

Attack Class	Precision	Recall	F1 Score	Support
DrDoS NTP	0.9982	0.9982	0.9982	3,281
TFTP	0.9983	0.9962	0.9973	66,555
DrDoS MSSQL	0.9840	0.9869	0.9855	14,097
DrDoS NetBIOS	0.9533	0.9988	0.9755	11,766
DrDoS SNMP	0.8540	0.8563	0.8552	16,348
Syn	0.9185	0.7946	0.8521	4,498
DrDoS DNS	0.8660	0.6535	0.7449	16,529
DrDoS UDP	0.5801	0.5439	0.5614	9,109
DrDoS LDAP	0.4291	0.6619	0.5207	6,377
DrDoS SSDP	0.4922	0.5216	0.5064	7,698
UDPLag	0.3229	0.6662	0.4350	758

As it can be seen in Table 2, there is a strong heterogeneity in the detection capabilities of the model applied to particular types of DDoS attacks. Other attacks like DrDoS NTP, TFTP, and MSSQL had almost a perfect precision, Recall, and F1 score, indicating that the attacks have very unique and consistent traffic characteristics that allow learning to the model. NetBIOS, SNMP and Syn were moderate in performance, so that though they can be generally identified, they are more usually associated with high variability that can sometimes lead to false classification. The lowest scores were made in LDAP, SSDP, DrDoS UDP, DrDoS DNS, and UDPLag, with the F1 scores significantly lower, thus indicating more misunderstanding with other types of attacks. These findings support a high degree of variability in the detectability of the attacks and emphasize that some types of DDoS pose inherent issues to the model, which can be explained by the overlapping behavioral profiles and the reduced discriminatory abilities.

Table 3. Misclassification count per attack category

Attack Class	Misclassified Samples
DrDoS NTP	6
TFTP	126
DrDoS MSSQL	189
DrDoS NetBIOS	138
DrDoS SNMP	2,344
Syn	930
DrDoS DNS	5,729
DrDoS UDP	4,148
DrDoS LDAP	2,157
DrDoS SSDP	3,685
UDPLag	253

Table 3 presents the misclassification counts for each of the eleven attack classes, providing further evidence of the performance diversity observed in Table 2. The misclassification counts follow a clear gradient that corresponds closely with the F1 scores:

- Low misclassification (≤ 200 samples): DrDoS NTP (6), TFTP (126), DrDoS MSSQL (189), and DrDoS NetBIOS (138). These classes exhibit minimal classification errors, confirming their highly separable and consistent traffic patterns.
- Moderate misclassification (200 – 2,500 samples): DrDoS SNMP (2,344) and Syn (930) show moderate error rates, consistent with their intermediate F1 scores, suggesting occasional feature overlap with other traffic types.
- High misclassification (> 2,500 samples): DrDoS DNS (5,729), DrDoS UDP (4,148), DrDoS LDAP (2,157), DrDoS SSDP (3,685), and UDPLag (253). While UDPLag has a relatively low absolute misclassification count due to its small support (758 samples), its proportion of misclassified samples (33.4%) is among the highest, indicating significant classification difficulty. The four other classes in this group exhibit both high absolute misclassification counts and elevated misclassification proportions, reflecting overlapping traffic characteristics that challenge the model's discriminative ability.

This graded pattern confirms that attacks with distinctive amplification-based signatures (NTP, TFTP, MSSQL) are reliably separable, whereas UDP-based, low-rate, and protocol-mimicking attacks (DNS, UDP, LDAP, SSDP, UDPLag) exhibit behavioral profiles that overlap with other classes or resemble benign traffic, resulting in substantially higher error rates.

Table 4. PR-AUC scores across attack classes

Attack Class	Misclassified Samples
DrDoS NTP	0.999
TFTP	0.998
DrDoS MSSQL	0.991
DrDoS NetBIOS	0.983
DrDoS SNMP	0.881
Syn	0.843
DrDoS DNS	0.722
DrDoS UDP	0.618
DrDoS LDAP	0.549
DrDoS SSDP	0.538
UDPLag	0.469

The precision-recall in Table 4 shows that there are great differences between the threshold modulation responses of each type of DDoS attack. Classes with high performance have large PR-AUC areas, signifying that its accuracy and recall level stay constant despite varying the decision threshold. This stability indicates that these attacks have characteristic and easily recognizable behaviour patterns, so that the model would be able to identify them with a high degree of discrimination over a wide range of operating environments. Contrarily, in poorly performing classes, the PR curves are considerably lower and shrinking very fast, which holds true to the increased sensitivity to class imbalance and overlapping feature distributions. Based on this reason, such classes are more vulnerable to a change of threshold, which is likely to cause false positives or false negatives. Taken together, these observations confirm the idea that PR-AUC is a strong metric of class-specific performance, which outlines attack types that can be sustained in the face of threshold change, with those that continue to be difficult to classify sensibly.

Table 5. Training time and prediction speed of XGBoost

Metric	Value
Training Time (seconds)	124
Prediction Speed (flows/sec)	58,000

As it is indicated in Table 5, the XGBoost model is highly efficient in terms of its computations, which makes it suitable in real-time or near-real-time DDoS detection. The model can be retrained/ updated on a regular basis with a training duration of 124 s and without incurring the prohibitive overhead of computing. More importantly, a prediction throughput of about 58,000 flows per second shows the ability of the model to handle high volume network traffic, which is a requirement in some settings like SDN controllers, intrusion detection systems, cloud systems and large enterprise networks. This capacity to throughput ensures that inbound flows are analyzed at almost real-time, thus enabling quick detection of malicious traffic and timely prevention of potential DDoS attacks. Overall, the fast training and high rate of inference justify the XGBoost balance between performance and computing efficiency and thus support its use in the operational cybersecurity setting.

Table 6. Per attack difficulty scores

Attack Class	F1 Score	Difficulty Score (ADS)
DrDoS NTP	0.9982	0.0018
TFTP	0.9973	0.0027
DrDoS MSSQL	0.9855	0.0145
DrDoS NetBIOS	0.9755	0.0245
DrDoS SNMP	0.8552	0.1448
Syn	0.8521	0.1479
DrDoS DNS	0.7449	0.2551
DrDoS UDP	0.5614	0.4386
DrDoS LDAP	0.5207	0.4793
DrDoS SSDP	0.5064	0.4936
UDPLag	0.4350	0.5650

Table 6 outlines the per-attack difficulty scores, highlighting a high degree of diversity in the detectability of various attack types of DDoS attacks. The weakest scores in difficulty, which are concordant with the highest F1 values, prove that the most easily classifiable ones are DrDoS NTP and TFTP because of their unique and recurring patterns of features. The classes of moderately difficulty, like the SNMP and Syn, are characterized by higher values of ADS, showing more overlap with other types of traffic and increased variability in the properties of the flow. The hardest categories of attacks, namely LDAP, SSDP, DrDoS UDP, and UDPLag, are classified as the most challenging since they have the highest difficulty scores, which could be attributed to the low F1 scores of the attacks.

The classes, probably, have structural similarities with benign or other attack traffic, thus making it difficult to differentiate benign and non-benign traffic using the model. On the whole, the ADS analysis proves that the detectability of DDoS differs greatly across categories, which highlights the importance of specific improvements that would be needed concerning the most challenging classes.

Table 7. Difficulty ranking

Rank	Attack Class	Difficulty
1	DrDoS NTP	Easiest
2	TFTP	
3	DrDoS MSSQL	
4	DrDoS NetBIOS	
5	DrDoS SNMP	Medium
6	Syn	
7	DrDoS DNS	Hard
8	DrDoS UDP	
9	DrDoS LDAP	
10	DrDoS SSDP	
11	UDPLag	Hardest

Table 7 provides a rank-based summary of the attack types, from the easiest to detect to the most difficult ones, based on the difficulty score they have computed. DrDoS NTP is considered to be the most manageable category, followed by TFTP, DrDoS MSSQL, and DrDoS NetBIOS, all having very distinct traffic patterns that the model can identify without much overlap. SNMP and Syn appear to be moderate in the intermediate range in that the model is usually accurate in identifying them, but their patterns of features sometimes overlap with other types of traffic. Categories that are the most powerful, starting with DrDoS DNS and continuing with DrDoS UDP, LDAP, SSDP, and finishing with UDPLag, are quite challenging to classify.

These types of attacks are also described to have more complex, noisy or inconsistent signatures, and hence they are more prone to misclassification. Taken together, the ranking supports previous findings that the detection difficulty varies significantly among classes, hence the need to develop specific mitigation measures that would address the most challenging types of attacks.

Figure 1 shows the F1-scores of the individual DDoS attack classes, which provides a ROC-like view of the discriminative ability of the model. Classes like DrDoS NTP, TFTP and MSSQL have the highest heights of the bar, indicating almost perfect separability and very reliable traffic patterns that the model identifies with a small error. In-between classes, such as NetBIOS, SNMP and Syn, are moderately detectable, whereas the height of the bars of classes such as DrDoS DNS, DrDoS UDP, LDAP, SSDP, and UDPLag are much lower, indicating a significant challenge due to a mix of overlapping or noise feature patterns. In general, the number highlights excessive differences in the complexity of classes and detection.

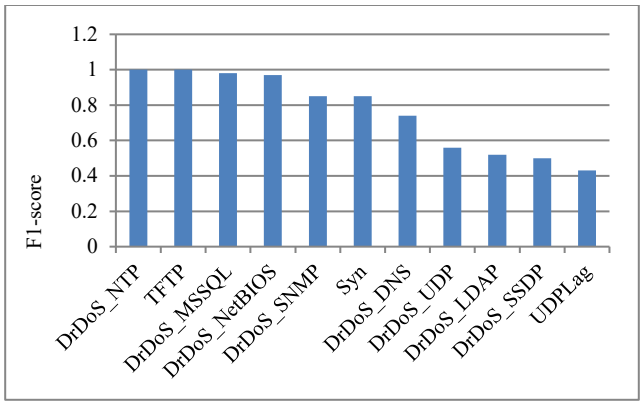

Fig 1. ROC-like interpretation per DDoS attack (F1-score proxy)

As shown in Figure 2, the Precision and Recall change differently with the various classes of DDoS attacks, thus illustrating the predictability of the model. Classes like DrDoS NTP, TFTP and MSSQL have high and very similar Precision and Recall values, thus showing that the model recognizes those attacks while subjecting other traffic to minimum mislabeling. In the middle of performance classes, such as NetBIOS, SNMP and Syn, there is a moderate variance between the Precision and Recall curves, and this shows some misclassifications. On the other hand, the lower curves of LDAP, SSDP, DrDoS UDP, DrDoS DNS and UDPLag indicate that Recall or Precision has decreased considerably, indicating the problem in differentiating these attacks as different. On the whole, this figure shows that there are some types of attacks, which have a similar performance at the levels of decisions, as well as those which have stronger sensitivity and can be erroneous.

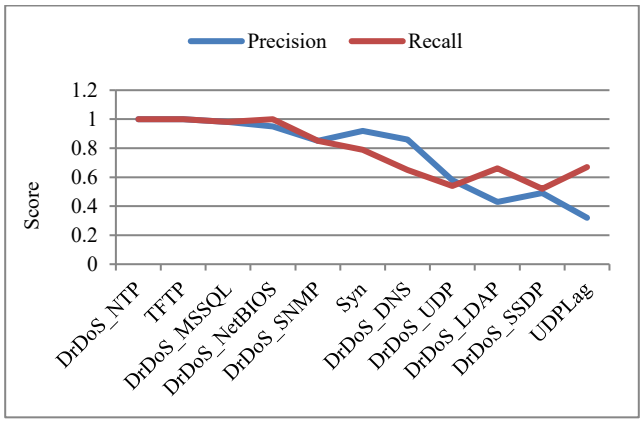

Fig. 2 Precision–Recall curves per class

Figure 3 shows the Attack Difficulty Scores (ADS) of each DDoS type and thus gives the visual prioritization of the difficulty of detection of each attack. Lower bars, such as the ones of attacks related to drDoS NTP, TFTP and MSSQL, signal that the attacks have rather distinct and uniform patterns which can easily be identified by the classifier. Class midrange scores in NetBIOS, SNMP and Syn are indicative of moderate difficulty as feature overlap in parts causes false classifications. The tallest bars, which are those that are related to LDAP, SSDP, DrDoS -UDP, DrDoS -DNS, and more so UDPLag, demonstrate huge problems in classification that are due to noisy, inconsistent, or highly similar traffic activity. In general, the figure reveals a significant disparity in detectability between the types of attacks and also presents categories that require additional model development.

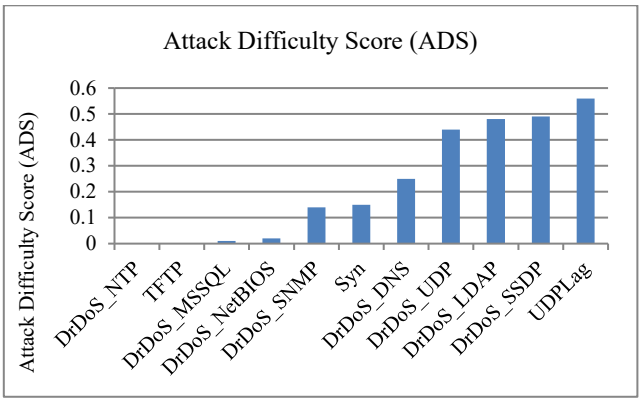

Fig. 3 The Attack Difficulty Scores (ADS)

Figure 4 represents a direct comparison of the precision, Recall, and F1-score of each type of DDoS attack, thus providing an overall evaluation of the model performance in the three key measures. Classes with high performance, like DrDoS NTP, TFTP and MSSQL, have consistent high bars in all measures, showing that they have a high accuracy, reliable detection, and low misclassification. Some classes like NetBIOS, SNMP and Syn have a moderately balanced value indicating that they have occasional inconsistencies in predictability. On the other hand, the bars of LDAP, SSDP, DrDoS UDP, DrDoS DNS, and UDPLag are much lower, which is explained by low accuracy, low recognition, or both. These tendencies verify considerable differences in detectability rates between types of attacks and identify the classes with the greatest contribution to aggregate model complexity.

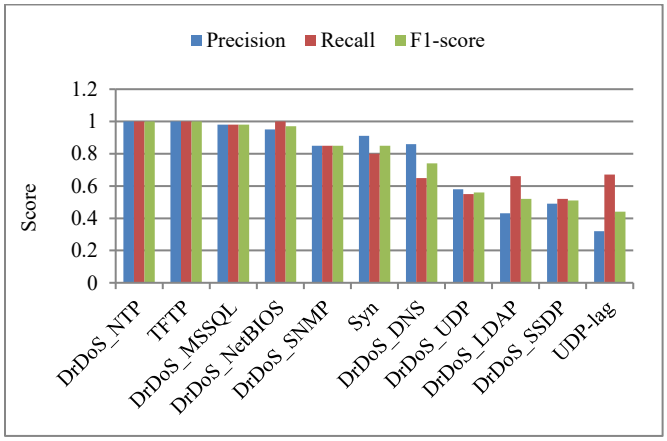

Fig. 4 Comparative performance metrics

In this study, it is shown that there is a significant difference in the detectability of the various types of DDoS attacks when an XGBoost multi-classifier is used. As the analysis reveals, some type of attack can be easily distinguished by different feature signatures; on the other hand, some have similar behavioural characteristics, making identification difficult. Classes with high scores, such as DrDoS NTP, TFTP and DrDoS MSSQL, had near-perfect recalls, F1 scores and precision, which showed the presence of clear patterns that can easily be discovered using gradient-boosted trees. These results correspond to the existing literature that amplification-based attacks have special packet-level and traffic rate properties, which can be learned by machine-learning models with low uncertainty [8, 16]. On the contrary, classes with lower scores, including DrDoS 0 DNS, DrDoS 0 UDP, LDAP, SSDP, and UDPLag, were more difficult to differentiate, and so were obtained more F1 scores with higher error rates. Such attacks often produce traffic patterns analogous to benign bursts or coinciding with other DDoS behaviors, which is consistent with the existing literature that structural similarity between some attack vectors can lead to the difficulty in detection by machine-learning-based intrusion-detection systems [17, 18].

The most challenging attack classes also help justify the fact of some inbuilt differences among the types of attacks as well. The Per Attack Difficulty Score (ADS) indicated that the classifications of the two hardest types of LDAP, SSDP, DrDoS UDP and UDPLag, share two common features: first of all, the high intra-class variance and secondly, the resemblance to the rest of the UDP-based flood behaviors. It has been confirmed by this observation that low-rate or protocol-mimicking attacks of DDoS integrate better with regular traffic, hence lowering the capacity of the classifier to draw consistent decision boundaries [23, 24]. Misclassification patterns also support the fact that multiple errors are made either between classes derived out of UDP and volumetric variants, or that finer-grained feature engineering or flow-level temporal modelling is needed. In comparison, the attacks that had a strong amplification and uniform amplification pattern had much fewer misclassifications, and so the hypothesis that clean protocol behaviors allow better machine-learning separation was confirmed.

The results of the Precision -Recall curve also indicate another crucial tendency: XGBoost will be persistent regarding the high-separability attacks, but the more problematic outcomes are observed when the distribution of the classes is biased, or the nature of the decision boundary is too narrow. It is also in line with previously conducted studies that indicate that PR-AUC is more sensitive to class rarity, which makes it a better predictor of multi-class intrusion detection with skewed data than ROC-AUC [19, 20]. The significantly lower PR-AUC scores in LDAP, SSDP and UDPLag suggest that these classes require more complex representations, which could be through sequence-based models, including Bi-LSTM or Transformer structures, which have been demonstrated to be effective in capturing tiny temporal changes in network flows [10, 25].

The findings of the computational efficiency imply that XGBoost can well fit real-time intrusion-detection systems; this algorithm is capable of handling a flow speed exceeding 58,000 messages per second, and it has high accuracy rates. The finding is consistent with previous studies that reveal that the best trade-off between speed and predictive accuracy of boosted-tree models in high-throughput network environments is achieved [21]. XGBoost is still practicable to more realistic operation applications like SDN controllers, edge devices or cloud-based monitoring nodes, yet deep-learning models with large absolute accuracy are significantly more expensive to compute, have much longer latency, and require limited resource usage.

It was found that the performance of the multi-class DDoS detection system is not only dependent on the algorithm used but also on the nature of each type of attack. The study proposes a diagnostic framework to evaluate the per-class difficulty, which allows the researchers and practitioners to understand which attacks need further feature engineering, data augmentation, or other learning structures. The results highlight the importance of being more adaptive and context-aware in regards to attack categories that are similar to legitimate traffic despite the potential of machine learning to provide a strong detection capability against the well-defined DDoS behaviors. The findings are part of a developing pattern where hybrid solutions are becoming more and more popular, such as signature-based heuristics and machine-learning classifiers or federated and adversarial learning to increase resilience to new threats [22]. According to the analysis, it finally proves that the multi-class DDoS detection is possible, but more optimization of the model setup, feature discovery, and heterogeneity of the datasets is required to achieve the best detection of all types of attacks.

3.3. Limitations and Generalization Considerations

Despite such strengths and fairly stable performance of the suggested framework when applied to a wide range of types of DDoS attacks, the current research has multiple limitations that limit the generalizability.

First, an experimental assessment was conducted on a single benchmark dataset, which uses an already defined collection of attack types and flow-level characteristics. Therefore, the performance metrics that will be reported herein mostly reflect the discriminatory capabilities of the model to the known types of DDoS, and not as much as its capability to detect completely new attack patterns or zero-day actions.

Secondly, the classifier was trained and tested on only attack-only traffic, and the benign traffic was not included in

the multi-class formulation. Therefore, the reported metrics are used to measure inter-attack discrimination instead of the true detection of the model in mixed traffic conditions. Although this method enables an analysis of per-attack difficulty, it does not allow conclusive results to be made on deployment readiness in heterogeneous traffic scenarios.

Third, cross-dataset generalizability was not determined, although the stratified cross-validation and repeated runs were introduced to ensure statistical stability. The network topology, traffic volume, feature-extraction pipelines or attack implementations could differ between datasets, which can affect the model performance. Therefore, the suggested Attack Difficulty Score cannot be viewed as something absolute but as dataset-relative.

In future studies, special focus will be put on testing the given framework on a scale of the datasets, including benign traffic, and on the ability to withstand unnoticed or changing attack patterns. Such extensions are expected to provide a more detailed assessment of the potential of generalization and feasibility in a practical operational intrusion detection system.

4. Conclusion

To better substantiate this, this paper is going to present evidence that models of DDoS attacks improved significantly when hybrid datasets are used, based on the approach of using a wide variety of sources to achieve this information. The multi-class classifier based on the XGBoost algorithm, with the use of an expanded and diversified dataset, was able to detect both major and minor differences between eleven types of DDoS attacks.

Experimental results demonstrated that some types of classes (DrDoS NTP, TFTP, MSSQL) were identified almost perfectly, and other ones (LDAP, SSDP, DrDoS UDP, UDPLag) were more difficult. These findings confirm that the heterogeneity of DDoS traffic is a major factor that defines the model performance, and per-class analysis is important in evaluating the system to a dot. The difficulty scoring, precision, recall analysis, confusion matrices, and computational efficiency measures are measures that holistically explain the strengths and weaknesses of the model. In most cases, it turned out that XGBoost was a highly desirable model when it comes to multi-class DDoS classification, with a good balance of accuracy, stability, and workability in real -/or very close to real-time detection environments. As noted in the study, the use of a wide scope of well-preprocessed data is valuable in enhancing generalization and reducing bias, especially when dealing with dynamic and challenging patterns of attacks. Future research could aim at attaining high detection rates of the most problematic attack types, optimization of adversarial training to be more robust, and testing of other deep-learning models, such as BiLSTM and implementation in a real-world setup, such as an SDN controller or cloud-edge defense. Such innovations have the promise of creating more resilient, versatile, and smart systems of cybersecurity, and thus improving resilience to the ever-evolving world of DDoS attacks.

4.1. Code Availability

The Python code used to process the data, choose features, train models, cross-validate and evaluate can be given to the respective author on reasonable request. The codebase contains fixed random seeds, described hyperparameter values, and scripts required to reproduce all experiments and figures in this paper. The repository will be released publicly after the acceptance of the manuscript.

Conflicts of Interest

The authors declare that there is no conflict of interest regarding the publication of this paper. All authors confirm that no financial, professional, or personal relationships influenced the conduct or outcomes of this research.

Funding Statement

This research received no specific grant from any funding agency, commercial entity, or not-for-profit organization. The authors undertook the study and publication independently without external financial support.

Acknowledgments

The authors express their appreciation to the faculty and research mentors of the State University of Northern Negros for their guidance during the development of this study. Both authors contributed equally to the research work and preparation of this manuscript.

References

[1] Donghwoon Kwon et al., "A Survey of Deep Learning-based Network Anomaly Detection," *Cluster Computing*, vol. 22, no. S1, pp. 949-961, 2017. [CrossRef] [Google Scholar] [Publisher Link]

[2] Nazrul Hoque, Dhruba K. Bhattacharyya, and Jugal K. Kalita, "Botnet in DDoS Attacks: Trends and Challenges," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 4, pp. 2242–2270, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[3] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization," *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP 2018)*, SCITEPRESS – Science and Technology Publications, vol. 1, pp. 108-116, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[4] Nour Moustafa, and Jill Slay, "The Evaluation of Network Anomaly Detection Systems: Statistical Analysis of the UNSW-NB15 Dataset," *Information Security Journal: A Global Perspective*, vol. 25, no. 1-3, pp. 18-31, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[5] Nickolaos Koroniotis et al., "Towards the Development of Realistic Botnet Dataset in The Internet of Things for Network Forensic Analytics: Bot-IoT Dataset," *Future Generation Computer Systems*, vol. 100, pp. 779-796, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[6] Rodrigo Braga, Edjard Mota, and Alexandre Passito, "Lightweight DDoS Flooding Attack Detection using NOX/OpenFlow," *IEEE Local Computer Network Conference*, Denver, CO, USA, pp. 408-415, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[7] Kun Wang et al., "Detection and Mitigation of DDoS Attacks based on Multi-Dimensional Characteristics in SDN," *Scientific Reports*, vol. 14, no. 1, pp. 1-19, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[8] Yuyang Zhou et al., "Building an Efficient Intrusion Detection System based on Feature Selection and Ensemble Classifier," *Computer Networks*, vol. 174, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[9] Hardik Arya et al., "Adaptive Sliding Window and LightGBM-based DDoS Attack Detection Framework for IoT Networks," *Peer-to-Peer Networking and Applications*, vol. 19, no. 1, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[10] Amal M. Al-Eryani, Fatma A. Omara, and Eman Hossny, "A Deep Learning GRU-BiLSTM for DDoS Attack Detection," *SN Computer Science*, vol. 6, no. 6, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[11] Ameur Salem Zaidoun, and Zied Lachiri, "A Hybrid Deep Learning Model for Multi-Class DDoS Detection in SDN Networks," *Annals of Telecommunications*, vol. 80, no. 5-6, pp. 459-472, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[12] S. Pradeesh, M. Jeyakarthic, and A. Thirumalairaj, "Enhanced Hybrid Approach for Multi-Class DDoS Attack Detection and Classification in Software-Defined Networks using Remote Sensing and Data Analytics," *Remote Sensing in Earth Systems Sciences*, vol. 8, no. 2, pp. 530-544, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[13] Hakan Aydin, and Muhammed Ali Aydin, "A Multi-Agent-based DdoS Detection and Defense System Design with Federated Learning and Blockchain in Public Cloud Network Environment," *Cluster Computing*, vol. 28, no. 16, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[14] Qasem Abu Al-Haija, and Ayat Droos, "Resilient Intrusion Detection System for Adversarial Attacks on Low-Rate DDoS," *International Journal of Machine Learning and Cybernetics*, vol. 16, no. 10, pp. 8473-8502, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[15] Rohan Doshi, Noah Apthorpe, and Nick Feamster, "Machine Learning DDoS Detection for Consumer Internet of Things Devices," *2018 IEEE Security and Privacy Workshops (SPW)*, San Francisco, CA, USA, pp. 29-35, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[16] Mohamed Amine Ferrag et al., "Deep Learning for Cyber Security Intrusion Detection: Approaches, Datasets, and Comparative Study," *Journal of Information Security and Applications*, vol. 50, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[17] Markus Ring et al., "A Survey of Network-based Intrusion Detection Data Sets," *Computers and Security*, vol. 86, pp. 147-167, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[18] Nour Moustafa, and Jill Slay, "UNSW-NB15: A Comprehensive Data Set for Network Intrusion Detection Systems (UNSW-NB15 Network Data Set)," *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, ACT, Australia, pp. 1-6, 2015. [CrossRef] [Google Scholar] [Publisher Link]

[19] Nathan Shone et al., "A Deep Learning Approach to Network Intrusion Detection," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 2, no. 1, pp. 41-50, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[20] Preeti Mishra et al., "A Detailed Investigation and Analysis of using Machine Learning Techniques for Intrusion Detection," *IEEE Communications Surveys and Tutorials*, vol. 21, no. 1, pp. 68-728, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[21] Anna L. Buczak, and Erhan Guven, "A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection," *IEEE Communications Surveys and Tutorials*, vol. 18, no. 2, pp. 1153-1176, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[22] Amardeep Kumar, Danish Ali Khan, and Ruhul Amin, "Detecting Application Layer DDoS Attacks with RSPF: A Hybrid Ensemble Learning Approach," *Cluster Computing*, vol. 28, no. 15, 2025. [CrossRef] [Google Scholar] [Publisher Link]

[23] Naziya Aslam, Shashank Srivastava, and M.M. Gore, "ONOS DDoS Defender: A Comparative Analysis of Existing DDoS Attack Datasets using Ensemble Approach," *Wireless Personal Communications*, vol. 133, no. 3, pp. 1805-1827, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[24] Min-Joo Kang, and Je-Won Kang, "Intrusion Detection System using Deep Neural Network for in-Vehicle Network Security," *PLOS ONE*, vol. 11, no. 6, pp. 1-17, 2016. [CrossRef] [Google Scholar] [Publisher Link]

[25] Razan Abdulhammed et al., "Features Dimensionality Reduction Approaches for Machine Learning-based Network Intrusion Detection," *Electronics*, vol. 8, no. 3, pp. 1-27, 2019. [CrossRef] [Google Scholar] [Publisher Link]