Tree-based Ensemble Algorithms and Feature Selection Method for Intelligent Distributed Denial of Service Attack Detection

Fauzi Adi Rafrastara[1], Guruh Fajar Shidik[1,*], Wildanil Ghozi[1], Nova Rijati[1] and Oki Setiono[2]

[1]*Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang, Indonesia*
[2]*Faculty of Health Science, Universitas Dian Nuswantoro, Semarang, Indonesia*
E-mail: guruh.shidik@dsn.dinus.ac.id
[*]*Corresponding Author*

Received 09 September 2024; Accepted 19 November 2024

Abstract

DDoS is one of hackers' mainstay weapons which can cause a decrease in network performance and damage servers. To overcome DDoS attacks, the challenge is to detect and block attacks simultaneously. Traditional classification methods are not effective at distinguishing between attack traffic and normal traffic. In this study, we introduce an ensemble-based machine learning algorithm, paired with an improved Gini index for feature selection, to detect DDoS attacks. Our approach used UNSW_NB15 dataset from Kaggle. Three tree-based ensemble algorithms are used in this research, namely Random Forest, XGBoost, and AdaBoost. By combining each ensemble algorithms with enhanced gini index, all those three algorithms outperformed

Journal of Cyber Security and Mobility, Vol. 14_1, 1–24.
doi: 10.13052/jcsm2245-1439.1411

the baseline models that used single decision tree classifier. XGBoost with gini index achieved the best result with 97.30% for accuracy, recall, and precision, and 96.90% for F1-score. This approach is able to improve the algorithm's performance while loweing the number of features.

Keywords: DDoS detection, ensemble learning, feature selection.

1 Introduction

In this modern era, warfare no longer only occurs on land, sea, and air, but can also occur in cyberspace [1]. Warfare in cyberspace is commonly referred to as cyber warfare, while attacks are referred to as cyberattacks. The emergence of cyberattacks is inseparable from the rapid and massive growth of the internet, which presents its own challenges in terms of security [2]. Therefore, the field of network security becomes very important to protect all business and industry activities, including financial transactions, email delivery, and online-based services [3]. One of the most feared types of cyberattacks that must be countered is the Distributed Denial of Service (DDoS) attack [4].

DDoS attacks continue to occur all the time in cyberspace, being one of the most frequently used and sophisticated attacks. Therefore, it is not surprising that DDoS is known to be very dangerous and difficult to overcome [2, 5, 6]. DDoS is one of the main weapons of hackers that can cause network performance degradation, paralysis of access to a website, and even damage to servers [7, 8]. DDoS can cause network traffic and prevent authorized users from accessing the service [9, 10]. DDoS can attack any computer or website whenever the attacker wants, even without the user realizing it. DDoS attacks can occur at the government level, companies, and even individuals [4].

In practice, hackers will flood the target website or computer with very large traffic, which is much larger than the data packets that can be accommodated by the targeted website or computer. Therefore, the website or computer becomes very burdened until it can no longer be accessed or used. The data packets used in DDoS are usually incoming messages, fake requests, or botnets. Once a website is attacked using DDoS, the website will slow down significantly, and the hacker will take over control of the website. When this happens, the admin can no longer access the website and control it [4]. The main goal of a DDoS attack is to disrupt or completely incapacitate the target system, often with the aim of extorting money from the victim. Therefore, it is not surprising that the majority of DDoS attacks are aimed at targets that attract the attention of these cybercriminals to launch their actions, such as

online shop websites, online games, banks, adult content, blogs and forums, and government websites [4].

In February 2020, as the largest cloud platform, Amazon Web Services faced a DDoS attack with an incoming traffic rate of 2.3 terabits per second (Tbps) that lasted for three consecutive days. However, AWS successfully mitigated the attack, preventing any severe disruption during this period. Furthermore, in 2021 the Bitcoin and cryptocurrency exchange EXMO was struck by a DDoS attack, causing its services to be down for five hours. GitHub also experienced a massive DDoS attack, where 126.9 million data packets were sent per second with a traffic rate of 1.3 Tbps [2].

To counteract DDoS attacks, the challenge is not only to detect the presence of attacks in real time, but also to be able to block the attacks at that very moment [11]. The most important step in dealing with DDoS attacks is to first detect their presence. Once detected, steps to stop the attack need to be executed. The primary focus of this research is on the detection phase. Statistical-based detection methods operate using prior knowledge of network flow patterns [12]. However, in today's environment, malicious network flows are increasingly dynamic and evolving, so the accessibility of features for training models becomes more limited [9, 13, 14]. Conventional machine learning performs effectively by applying rules to a limited dataset [9]. This methods have performance limitations in distinguishing between traffic that contains attacks and normal traffic [11]. In this proposal, the researcher proposes a framework for detecting DDoS attacks by utilizing ensemble-based algorithms and feature selection methods to improve its performance.

2 Related Works

In the research on DDoS attack detection, Azmi et al. employed information gain for feature selection and several classification algorithms for comparative analysis, including Naive Bayes (NB), Decision Table, and Artificial Neural Network (ANN). The dataset used is the UNSW-NB 15 downloaded from kaggle.com. The Decision Table algorithm yielded the highest accuracy performance, scoring 88.43% [4].

Ismail et al. introduce a systematic approach for detecting DDoS attack. Initially, they chose the dataset from a GitHub repository, namely UNSW-NB15 which includes DDoS attack data. They utilized Python and Jupyter notebook for data pre-processing. Next, the dataset was divided into two groups: the independent class and the dependent class. They used the Random Forest (RF) and XGBoost ensemble-based classification methods after

normalizing the data. They discovered that the RF achieved roughly 89% accuracy for Precision and Recall in the initial classification. Furthermore, the proposed model achieved an average accuracy of approximately 89%, which is very good and astounding. It is crucial to remember that the F1-score is represented by the average Accuracy, which is 89%. They found that precision and recall for XGBoost, the second classifier, were roughly 90% correct. For the recommended model, they saw an accuracy of almost 90%, which is amazing and really exceptional. The accuracy in this model denotes a 90% F1-score [15].

Bouke et al. used the Gini index feature selection method to create a tree-based model for intelligent DDoS attack detection. This study used UNSW-NB15. A total of 1,140,045 instances were selected from part 3 and part 4 of this dataset. With an overall accuracy of 98%, their method surpassed benchmark models that made use of more sophisticated algorithms, like RF and XGBoost. With their improved Gini Index feature selection technique, they were able to choose only 13 security features out of 45. This significantly reduces the dimensionality of the data and helps prevent overfitting issues. Furthermore, their approach incorrectly classifies only 2% of test situations, which lowers the false alert rate [16].

Yuhua et al. proposed an integrated feature selection approach for MLP-based intrusion detection systems, referred to as IGRF-RFE (Information Gain and Random Forest – Recursive Feature Elimination). IGRF-RFE serves as the initial step in the feature selection process. It aims to reduce the dimensionality of the features. After IGRF-RFE is applied to remove features that negatively affect the performance of MLP. The Authors used a Multilayer Perceptron (MLP) with two hidden layers, both for RFE and as the final classifier. The research utilized the UNSW-NB14 dataset. As a result, the proposed model has reduced the number of feature from 42 to 23. The detection accuracy achieved was 84.24%, which outperformed the experiment without feature selection (which had an accuracy of 82.25%) [17].

On other study, Alqarni conducted a majority vote-based ensemble approach to detect DDoS attacks in cloud computing. He utilized four conventional classifier algorithms, and the outputs of all classifiers were combined to make a decision. The conventional algorithms used include Naive Bayes (NB), Decision Trees (DT), Support Vector Machines (SVM), and k-Nearest Neighbors (kNN). This research employed the CICD-DOS2019 dataset, which contains 80 original features that were reduced to 15 features using the chi-squared feature selection algorithm. This research achieved the

best performance with the ensemble method, resulting in an accuracy score of 98.02%, a sensitivity of 97.45%, and a specificity of 98.65% [18].

3 Research Methods

The DDoS attack detection system proposed in this research, which uses ensemble machine learning algorithms and gini index, is designed to automatically identify potential DDoS and other attacks, and then notify system administrators. The design of the proposed model is depicted in Figure 1. This section will also cover the research methodology which involves several steps in developing the system.

3.1 Data Collection

In every machine learning project, the initial step in the implementation process involves dataset preparation. As per reference [16], the majority of machine learning methods require explicit data formatting, implying that datasets typically need some level of preparation to extract meaningful information. Comprehensive data preparation allows analysts to have confidence in their data, comprehend it better, and pose more insightful queries, thereby enhancing the accuracy and depth of their analyses.

The dataset utilized in this research is UNSW-NB15 (downloaded from kaggle.com). This dataset was developed by the Australian Center for Cyber Security (ACCS) in collaboration with various researchers globally [19]. This dataset includes a compilation of contemporary network traffic, encompassing both normal and attack instances. This dataset consists of 4 files with details shown in Table 1.

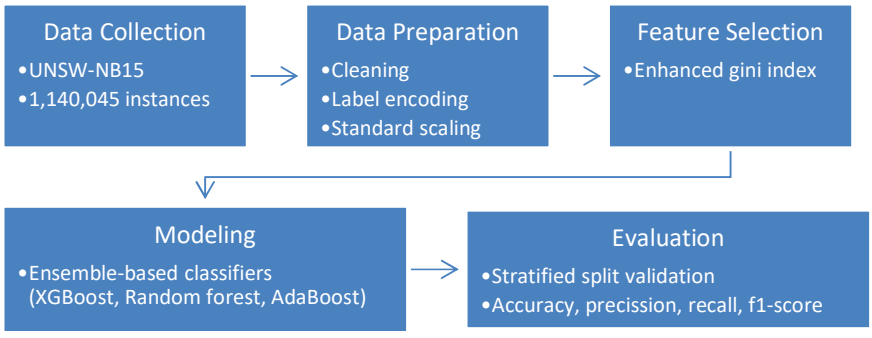

Figure 1 Model architecture overview.

Table 1 The dataset CSV files

File Name	Column	Target Class	Instances
UNSW-NB15_1	49	10	700,001
UNSW-NB15_2	49	10	700,001
UNSW-NB15_3	49	10	700,001
UNSW-NB15_4	49	10	440,044

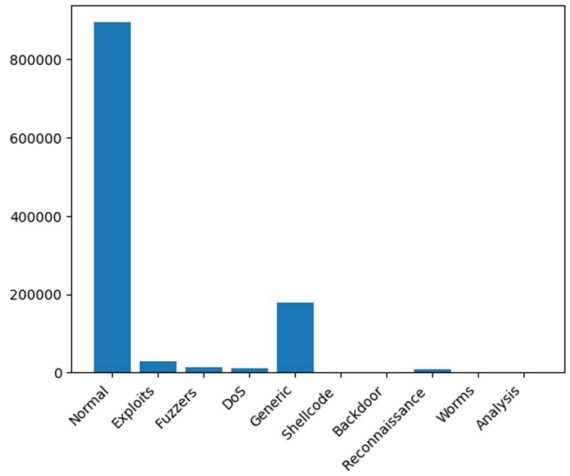

Figure 2 Class distribution of dataset.

To examine the proposed architecture, the dataset files used in this research are only UNSW-NB15_3 and USNW-NB15_4 as in the experiments in paper [16]. The total number of records resulting from the combination of the two dataset files is 1,140,045 records. The number of features in this dataset is 49 columns. The data types in the dataset are also diverse, it consists of nominal, integer, float, time-stamp, and binary. The target variable we use is attack-cat, which contains ten class labels, with all records distributed across each class as shown in Figure 2. The class distribution in this dataset as shown in Figure 2, is skewed. This imbalance can lead to biased classification results, particularly for the minority class [20].

3.2 Data Preparation

In the realm of data science, it is often stated that approximately 50% to 80% of a data science project is dedicated to data preparation [21]. This critical phase involves cleaning, transforming, and organizing raw data into a format

suitable for analysis. The reason this phase is so time-intensive is due to the inherent complexity and disorderliness of real-world data. Despite being a demanding task, data preparation is of utmost importance as the quality of the data directly influences the accuracy of the subsequent analysis and predictions [15, 16]. The default steps in data preparation typically include label encoding, scaling, and feature selection [16]. Label encoding is used to convert categorical data into a format that can be understood by machine learning algorithms. Scaling is applied to standardize the range of input features, ensuring that no particular feature dominates the others. Lastly, feature selection is the process of choosing the most important features in your data that contribute most to the output variable or result [22]. These steps are crucial in preparing high-quality data for effective analysis. Before we perform those three steps, firstly we do the data cleaning by removing several irrelevant features in the dataset [23].

3.2.1 Data cleaning

Data cleaning involves detecting and addressing errors, inaccuracies, and inconsistencies within datasets to ensure their accuracy and reliability [14, 24]. This process is crucial in maintaining the quality and reliability of data, which directly impacts the performance of data analysis and machine learning models. Data cleaning can involve various tasks such as handling missing values, removing duplicates, correcting inconsistent entries, and dealing with outliers [25]. For Instance, missing values can be addressed using methods such as deletion, imputation, or prediction. Duplicates can be detected and eliminated to prevent redundancy. Inconsistent entries, such as differences in spelling or formatting, can be standardized to ensure uniformity [21]. In this step, we found no missing values in the dataset, no duplication and inconsistent entries, and we removed 8 irrelevant features as mentioned in subchapter A. Data Collection.

3.2.2 Label encoding

Label encoding is a fundamental process in data science. It comes into play when dealing with categorical data [26]. Many datasets contain features that are categorical, representing distinct categories or classes. However, most machine learning algorithms require numerical input. Label encoding is a process that converts categorical values into numerical form [27, 28]. For instance, ‘red’ might become 0, ‘blue’ might become 1, and ‘green’ might become 2. This allows machine learning algorithms to process the data while preserving the meaningful representation of the categories. Another

alternative method to deal with the categorical data is one-hot encoding. However, one-hot encoding has its drawbacks. The main one is the increase in the dimensionality of the data [16, 29]. If a categorical variable has many categories, one-hot encoding can lead to a large number of input features, leading to memory and computational challenges for machine learning models [28]. This problem is often referred to as the "curse of dimensionality" [30]. To avoid the high dimensionality

3.2.3 Scaling

Feature scaling is defined as a preprocessing step in machine learning that adjusts the range of feature values to ensure equal contribution from all features, irrespective of their original scales [31]. Feature scaling can improve the performance of many machine learning algorithms by helping them converge more quickly and accurately. Two common methods of feature scaling are normalization and standardization [32].

Standardization is a technique used in feature scaling that adjusts the value of features, so they have a mean of 0 and a standard deviation of 1. This is like transforming the scores of an exam into a "curve," where the average score is set to 0, and the spread of scores is set to 1. In machine learning, standardization guarantees that each feature has an equal impact on the model, irrespective of their original scale or units. Standardization can be done using Equation (1), where, z is the new scaled value, μ and σ are the mean and standard deviation of the features. In this research, the method of feature scaling employed is standardization.

$$z = \frac{x - \mu}{\sigma} \tag{1}$$

3.3 Feature Selection

Feature selection is a fundamental step in the realm of data mining, particularly in the context of DDoS attack detection. It involves identifying and selecting the most relevant features from a dataset, thereby reducing its dimensionality. This process is crucial for several reasons: Improved Model Performance; Reduced Computational Cost; Enhanced Model Interpretability [33].

The Gini Index is a widely used metric in machine learning, including feature selection. It quantifies the impurity of a dataset, with lower Gini Index values indicating higher purity. In the context of DDoS attack detection, the Gini Index is used to assess the discriminatory power of each feature [34].

The formula of Gini Index as shown in Equation (2).

$$Gini = 1 - \sum_{i=1}^{n} (p_i)^2 \tag{2}$$

Where, pi is the probability of an object being classified to a particular class. This research used the enhanced gini index feature selection method as proposed in [16]. This method identified 13 of the most significant features in the UNSW-NB dataset.

3.4 Modelling

In this stage, we employed split validation to separate data for learning process and testing the model [35]. The training dataset is utilized in the learning process to develop the model, enabling it to learn and make predictions. Conversely, the testing dataset is employed to evaluate the model's performance, providing an unbiased assessment of how well the model handles new, unseen data. The experiment employs a 70:30 ratios for training to testing data. After splitting the sample, the modelling process can commence. This research utilizes an ensemble classification algorithm for the proposed model.

In data mining, the ensemble method is an effective technique that merges several models to enhance predictive accuracy. It is based on the idea that combining multiple weak learners can create a stronger, more robust learner [36]. This approach is especially beneficial for handling large and complex datasets, as it helps to minimize both bias and variance [37]. The ensemble method has been effectively utilized across diverse fields such as healthcare, finance, and marketing. It has demonstrated its value as a powerful tool in data mining, offering reliable and precise predictions [38].

In this research, 3 ensemble-based classification algorithms were used, namely Random Forest, XGBoost, and AdaBoost. The Random Forest algorithm is a form of ensemble learning that builds multiple decision trees during training and outputs the class that is the mode of the classes for classification or the mean prediction for regression from these trees [39]. It operates on the bagging principle, where several subsets of the original dataset are generated, each subset is used to train a decision tree, and the final output is determined by majority voting for classification or averaging for regression. The Gini Index and Entropy are crucial concepts in decision tree construction, which are foundational to Random Forests. The Gini Index, also known as Gini Impurity, measures the probability of incorrectly classifying a randomly chosen instance. It quantifies the level of impurity or disorder

in a node of the decision tree. The objective is to minimize the Gini Index to achieve a purer node where all instances belong to a single class. The formula for the Gini Index is shown in Equation (2). Entropy, on the other hand, is a criterion that guides the decision on how a node should split based on the likelihood of a specific outcome, by examining all potential branches. This is represented mathematically in Equation (3). The goal is to minimize entropy, which would result in a more organized and efficient decision tree within the Random Forest. This process aids in creating more accurate and reliable predictions. Like the Gini Index, entropy helps to build an efficient decision tree by selecting the best splitter. The entropy of a split is always between 0 and 1, with 0 representing a pure (homogeneous) node and 1 representing a node with maximum disorder.

$$Entropy = \sum_{i}^{c} -p_1 \times \log_2(p_i) \tag{3}$$

eXtreme Gradient Boosting, or XGBoost for short, is an implementation of gradient boosting machines. These machines are made to choose a function that points in the direction of the negative gradient iteratively in order to maximize a cost function across function space. A parallel tree boosting method called XGBoost makes it possible to quickly and accurately handle a wide range of data science issues. It is compatible with multiple objective functions, such as ranking, classification, and regression. Scalability is one of XGBoost's main advantages in every situation. The system scales to billions of examples in distributed or memory-limited situations and operates more than ten times quicker on a single machine than popular alternatives currently in use [40].

Adaptive Boosting, or AdaBoost, is a technique that builds a powerful learner by combining several weak learners. Usually decision trees with a single split, or decision stumps, are these weak learners [41]. By giving examples that were erroneously classified in the previous iteration larger weights, the technique trains these weak learners iteratively on different data distributions. A weighted majority vote is used to make the final prediction, which takes into account each weak learner's accuracy.

3.5 Evaluation

Although many performance metrics have been suggested in the literature to assess classifier effectiveness, precision and recall are the most commonly used for evaluating classifiers on imbalanced data [42]. The confusion matrix,

Table 2 Confusion matrix

		Predicted	
		Positive	*Negative*
Actual	*Positive*	TP	FN
	Negative	FP	TN

a two-dimensional matrix, offers a comprehensive view of how a classifier predicts new samples. The rows represent the actual classes of objects, while the columns indicate the predicted classes by the classifier. Each cell in the matrix is categorized into True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). The confusion matrix is used to evaluate the performance of a classification model by calculating metrics such as accuracy, precision, recall, and F1-score [43]. The table of confusion matrix is presented in Table 2.

Accuracy is the proportion of accurately predicted occurrences to all instances in the dataset. It is a useful measure when the target variable classes in the data are nearly balanced. Mathematically, it is represented in Equation (4).

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{4}$$

Precision is the ratio of accurately predicted positive instances to all expected positives. A significant percentage of false positives is indicated by a low precision. It can be described by the formula in Equation (5).

$$Precission = \frac{TP}{(TP + FP)} \tag{5}$$

Recall, sometimes referred to as sensitivity, is the ratio of accurately predicted positive cases to the total number of actual positives. A large percentage of false negatives is indicated by a low recall. The formula for it is given by Equation (6).

$$Recall = \frac{TP}{(TP + FN)} \tag{6}$$

The F1-score is the harmonic mean of precision and recall [44]. It tries to find the balance between precision and recall. It is particularly useful when the class distribution is unbalanced. Equation (7) provides the formulaic

representation of it.

$$F1\text{-}score = \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

4 Experiment and Discussion

The experiment was conducted on a computer equipped with a Core i5-1200 processor, 16 GB of RAM, a 256 GB NVME SSD, and the Windows 11 operating system. The worksheet uses a Jupyter notebook with the Python programming language.

4.1 Data Preparation

Our proposed model is designed and assessed using the most recent DDoS attack dataset, which is categorized into 9 distinct attack types. We used 1,140,045 sample from the entire UNSW-NB15_3 and UNSW-NB15_4 dataset to reduce the computational load as done in previous studies [16]. The merged dataset initially comprised 49 columns. Through a manual feature selection process, irrelevant features such as 'label', 'srcip', 'dstip', 'sport', 'dsport', 'Stime', and 'Ltime' were eliminated. The 'Attack_cat' column was designated as the target variable for prediction. In this step, we found no missing values in the dataset, no duplication and inconsistent entries.

Given that the UNSW-NB15 dataset contains several nominal features, encoding was necessary to transform them into numerical representations. Label encoding was chosen to preserve the dimensionality of the feature space and mitigate computational overhead. The nominal features 'Proto', 'State', and 'Service' were encoded using this technique.

All features in the dataset were then numerical. However, to address the varying scales of these features, standardization was applied. This process involves transforming the features to have a mean of 0 and a standard deviation of 1.

4.2 Selected Feature

Many feature selection methods have been utilized in various studies. The challenge lies in selecting important features while discarding those that provide misleading information. We proposed a model that uses a Gini index for select the most importance feature to decrease data dimensionality, thereby increasing accuracy and speeding up the detection process. Feature selection

Table 3 Selected features

No	Name	No	Name	No	Name
1.	ct_dst_sport_ltm	6	ct_src_ltm	11	sintpkt
2.	ct_src_dport_ltm	7	ct_srv_dst	12	sttl
3.	ct_dst_src_ltm	8	ct_srv_src	13	Dintpkt
4.	Sload	9	is_sm_ips_ports		
5.	ct_state_ttl	10	ct_dst_ltm		

has been shown to significantly improve classification problems. We referred to previous research [16] to select the 13 most important security features for building an ensemble-based intrusion detection model (Table 3). The 13 features were obtained from a Gini Index-based ranking with a threshold of 0.15.

We then utilized these 13 features to compare our method with the approach proposed by [16].

4.3 Result and Comparison

We conducted a re-experiment on the approach proposed by [16]. They claimed that their approach can achieve 98% accuracy, precision, recall, and F1-score. They compared it with the method proposed by [15], and it is significantly better than the result from [15] that only produce 90% as the best score for precision, recall and F1-score. With the same dataset size and features mentioned in [16], we conducted a re-experiment with the decision tree algorithm as used in [16] and found the result to be 96.9% for accuracy and recall, and 96.70% for Precision and F1-score.

Our proposed model was evaluated using split validation, with 70% of the 1,140,045 UNSW-NB15 dataset allocated for training and the remaining 30% for testing. To enhance the relationship between training and testing data, stratified random sampling was performed five times, leading to improved evaluation results. Three ensemble-based classification algorithms we proposed in this study achieve better performance than previous related studies as shown in Table 4. To show a better comparison, we show the graph of our proposed method with previous related studies in Figure 3.

XGBoost performed the best with 97.30% accuracy, precision, and recall, along with a 96.90% F-score. Random Forest and AdaBoost achieved nearly identical scores, except in precision, where Random Forest (97.20%) slightly outperformed AdaBoost (97.10%). Both algorithms achieved 97.20% accuracy and recall, as well as a 96.90% F-score. Additionally, we compared our

Table 4 Experiment result and comparison on multiclass classification

Algorithm	FS Method	Features	Accuracy	Precision	Recall	F1-score
XGBoost *(Proposed)*	Gini Index	13	97.30%	97.30%	97.30%	96.90%
Random Forest	Gini Index	13	97.20%	97.20%	97.20%	96.90%
AdaBoost	Gini Index	13	97.20%	97.10%	97.20%	96.90%
Decision Tree [16]	Gini Index	13	96.90%	96.70%	96.70%	96.70%
XGBoost [15]	None	45	90.00%	90.00%	90.00%	90.00%
MLP [17]	IGRF-RFE	23	84.24%	83.60%	84.24%	82.85%

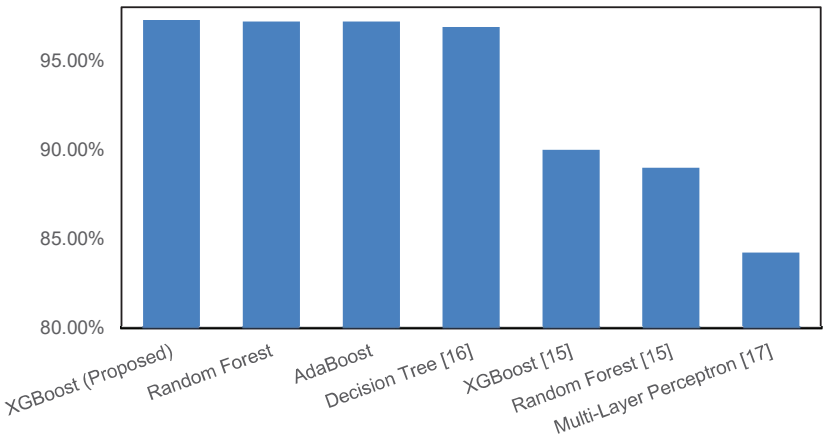

Figure 3 Accuracy comparison on multi class classification.

model with other state-of-the-art approaches proposed by [15, 17]. Despite this comparison, our model remained the top performer. Based on these results, we recommend using the Enhanced Gini Index feature selection method and XGBoost classification algorithms for this study.

The test results of our proposed model, shown in Table 5, further illustrate its performance across all attack categories. These results indicate that our proposed model is capable of accurately predicting DDoS attacks with high sensitivity and precision. Thus, the feature selection method combined with XGBoost is able to increase computational speed and efficiency in detecting DDoS attacks.

In the precision metric, our proposed model is able to identifying "Generic" attacks well with 99.70% precision and then the normal category of 99.10% precision indicates the ability to identify normal packets that are not DDoS attacks. However, our proposed model produces low precision in

Table 5 Result of all attack categories in XGBoost model

Attack Categories	Precision	Recall	F1-score
Normal	99.10%	99.60%	99.40%
Generic	99.70%	98.90%	99.30%
Exploits	59.40%	90.70%	71.80%
Fuzzers	69.60%	50.20%	58.40%
DoS	55.70%	6.60%	11.80%
Reconnaissance	89.70%	74.30%	81.30%
Analysis	75.20%	10.50%	18.50%
Backdoor	89.10%	10.40%	18.60%
Shellcode	50.50%	55.70%	53.00%
Worms	59.70%	44.90%	51.30%

"Shellcode" attacks with a precision of 50.50%, which means that the false positives in predicting these attacks are high enough to cause false alerts to system administrators.

Recall metrics show that our proposed model is very sensitive to "normal" packets that are not DDoS attacks with a recall of 99.60%. This indicates a low false negative value when categorizing normal packets so that normal computer network access is easily detected and does not cause a decrease in availability.

Confusion matrices are valuable tools for assessing the performance of classification models, as they display the number of correctly and incorrectly predicted instances. They allow for the measurement of classification performance discussed earlier. The confusion matrix in Figure 4 shows that the ensemble-based algorithm combined with the feature selection method is most effective at identifying "normal" packets rather than detecting attacks. This is also in accordance with the precision calculation in the previous discussion. We also conclude that the model we propose more often predicts "DoS" attacks as "exploits", this can still be tolerated because both are DDoS attacks so an alarm will be sent to the system administrator. What should be of concern is the "Fuzzers" attack prediction ability which is more often predicted as a "normal" packets. However, the confusion matrix shows that the model we developed exhibits high accuracy. Out of 1,710,070 instances tested, approximately 46,000 were incorrectly classified, resulting in an accuracy rate of 97.3%.

A high false negative rate in shellcode and fuzzer detection indicates a vulnerability in the system's defense mechanisms. The failure to detect these attacks can compromise the system's integrity and availability. Fuzzers

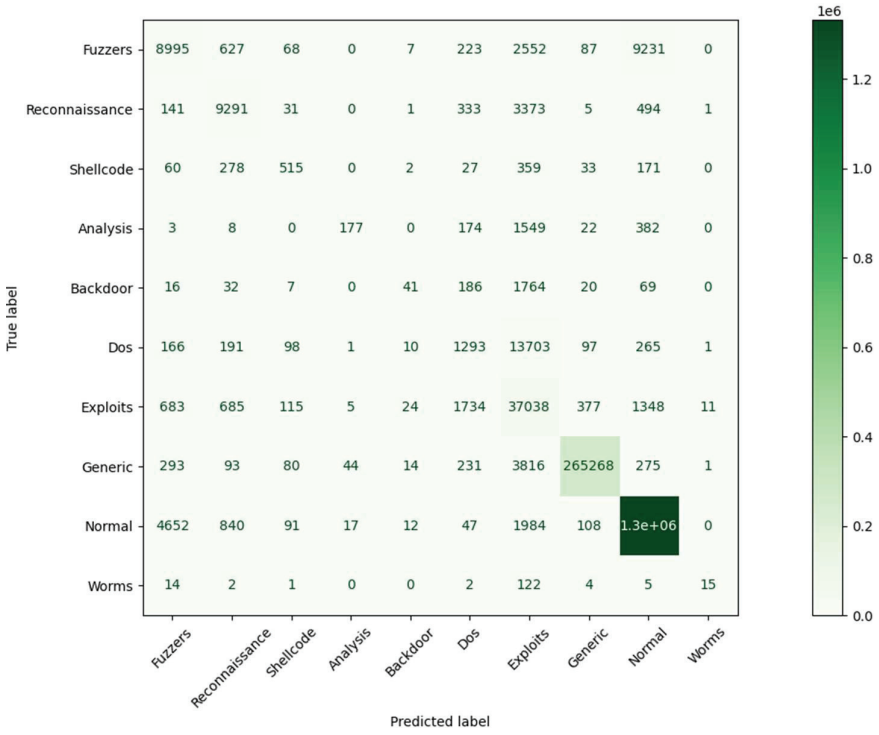

Figure 4 Confusion matrix of XGBoost with enhanced gini index.

can exploit system vulnerabilities to cause denial-of-service conditions, while shellcode can be used to execute malicious code that threatens data confidentiality.

The low F1-scores obtained for several attack categories, including Fuzzers, DoS, Analysis, Backdoor, Shellcode, and Worms as shown in Table 5, indicate the need for further investigation. A closer examination of the data distribution in Figure 2 reveals that these classes exhibit imbalanced data distribution. This is likely contributing to the high false positive rates observed in these classes. Additionally, the model's accuracy may be biased due to the dominance of the normal class. Therefore, in practical applications, this model can be adapted for anomaly detection, where the system identifies and permits "normal" network traffic. For future research, it is recommended to address the class imbalance issue by generating a new dataset or applying oversampling techniques.

5 Conclusion

To defend against DDoS attacks, it is essential to implement a system that can quickly and effectively distinguish between harmful and legitimate traffic. This study introduces a systematic method for detecting DDoS attacks. We began by selecting the UNSW_NB15 dataset from the Kaggle repository, which includes 1,140,045 traffic samples containing instances of DDoS attacks. This dataset was provided by the Australian Centre for Cyber Security (ACCS). We then conducted pre-processing task by performing cleaning, encoding, and scaling. Next, we employed three ensemble-based classification algorithms, specifically Random Forest, AdaBoost and XGBoost. The algorithms were enhanced by integrating them with a Gini index, a statistical measure of inequality. This integration resulted in a significant improvement in performance over the baseline models, which utilized a decision tree as a classifier. Among the three approaches, XGBoost, in conjunction with the Gini index, provided the highest performance, with an accuracy, recall, and precision of 97.30% and an F-score of 96.90%. This approach not only increased the performance of the algorithm but also effectively reduced the number of features required, thereby optimizing the computational efficiency.

Acknowledgment

We would like to express our gratitude to the Institute of Research and Community Service (LPPM), Intelligent Distributed Surveillance and Security (IDSS) research group, and the Faculty of Computer Science, Universitas Dian Nuswantoro for their support in facilities and funding for this research.

References

[1] F. Cremer, B. Sheehan, M. Mullins, M. Fortmann, B. J. Ryan, and S. Materne, "On the insurability of cyber warfare: An investigation into the German cyber insurance market," *Computers & Security*, vol. 142, p. 103886, Jul. 2024, doi: 10.1016/j.cose.2024.103886.

[2] R. Mall, K. Abhishek, M. S., A. Shankar, and A. Kumar, "Stacking ensemble approach for DDoS attack detection in software-defined cyber–physical systems," *Computers and Electrical Engineering*, vol. 107, p. 108635, Apr. 2023, doi: 10.1016/j.compeleceng.2023.108635.

[3] M. Alkasassbeh, G. Al-Naymat, A. B. A. Hassanat, and M. Almseidin, "Detecting Distributed Denial of Service Attacks Using Data Mining Techniques," *International Journal of Advanced Computer Science and Applications*, vol. 7, no. 1, 2016, doi: 10.14569/IJACSA.2016.070159.

[4] M. A. H. Azmi, C. F. M. Foozy, K. A. M. Sukri, N. A. Abdullah, I. R. A. Hamid, and H. Amnur, "Feature Selection Approach to Detect DDoS Attack Using Machine Learning Algorithms," *JOIV: International Journal on Informatics Visualization*, vol. 5, no. 4, p. 395, Dec. 2021, doi: 10.30630/joiv.5.4.734.

[5] M. Arunadevi and V. Sathya, "DDoS Attack Detection using Back Propagation Neural Network Optimized by Bacterial Colony Optimization," *IJIES*, vol. 16, no. 5, pp. 301–312, Oct. 2023, doi: 10.22266/ijies2023.1031.26.

[6] Z. Liu, Y. Wang, F. Feng, Y. Liu, Z. Li, and Y. Shan, "A DDoS Detection Method Based on Feature Engineering and Machine Learning in Software-Defined Networks," *Sensors*, vol. 23, no. 13, p. 6176, Jul. 2023, doi: 10.3390/s23136176.

[7] M. Khare and R. Oak, "Real-Time Distributed Denial-of-Service (DDoS) Attack Detection Using Decision Trees for Server Performance Maintenance," in *Performance Management of Integrated Systems and its Applications in Software Engineering*, M. Pant, T. K. Sharma, S. Basterrech, and C. Banerjee, Eds., Singapore: Springer Singapore, 2020, pp. 1–9. doi: 10.1007/978-981-13-8253-6_1.

[8] U. S. Chanu, K. J. Singh, and Y. J. Chanu, "A dynamic feature selection technique to detect DDoS attack," *Journal of Information Security and Applications*, vol. 74, p. 103445, May 2023, doi: 10.1016/j.jisa.2023.103445.

[9] M. Mittal, K. Kumar, and S. Behal, "Deep learning approaches for detecting DDoS attacks: a systematic review," *Soft Comput*, vol. 27, no. 18, pp. 13039–13075, Sep. 2023, doi: 10.1007/s00500-021-06608-1.

[10] Y. B. Sanap and P. Aher, "A Comprehensive Survey On Detection And Mitigation Of DDoS Attacks Enabled With Deep Learning Techniques In Cloud Computing," in *2023 6th International Conference on Advances in Science and Technology (ICAST)*, 2023, pp. 149–154. doi: 10.1109/ICAST59062.2023.10454990.

[11] J. Gera and B. P. Battula, "Detection of spoofed and non-spoofed DDoS attacks and discriminating them from flash crowds," *EURASIP Journal*

on Information Security, vol. 2018, no. 1, p. 9, Dec. 2018, doi: 10.1186/s13635-018-0079-6.

[12] F. O. Catak and A. F. Mustacoglu, "Distributed denial of service attack detection using autoencoder and deep neural networks," *IFS*, vol. 37, no. 3, pp. 3969–3979, Oct. 2019, doi: 10.3233/JIFS-190159.

[13] A. Bhardwaj, V. Mangat, R. Vig, S. Halder, and M. Conti, "Distributed denial of service attacks in cloud: State-of-the-art of scientific and commercial solutions," *Computer Science Review*, vol. 39, p. 100332, 2021, doi: https://doi.org/10.1016/j.cosrev.2020.100332.

[14] Q. Li *et al.*, "A comprehensive survey on DDoS defense systems: New trends and challenges," *Computer Networks*, vol. 233, p. 109895, 2023, doi: https://doi.org/10.1016/j.comnet.2023.109895.

[15] Ismail *et al.*, "A Machine Learning-Based Classification and Prediction Technique for DDoS Attacks," *IEEE Access*, vol. 10, pp. 21443–21454, 2022, doi: 10.1109/ACCESS.2022.3152577.

[16] M. A. Bouke, A. Abdullah, S. H. ALshatebi, M. T. Abdullah, and H. E. Atigh, "An intelligent DDoS attack detection tree-based model using Gini index feature selection method," *Microprocessors and Microsystems*, vol. 98, p. 104823, Apr. 2023, doi: 10.1016/j.micpro.2023.104823.

[17] Y. Yin et al., "IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset," *J Big Data*, vol. 10, no. 1, p. 15, Feb. 2023, doi: 10.1186/s40537-023-00694-8.

[18] A. A. Alqarni, "Majority Vote-Based Ensemble Approach for Distributed Denial of Service Attack Detection in Cloud Computing," *JCSANDM*, Mar. 2022, doi: 10.13052/jcsm2245-1439.1126.

[19] N. Moustafa and J. Slay, "UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference (MilCIS)*, Canberra, Australia: IEEE, Nov. 2015, pp. 1–6. doi: 10.1109/MilCIS.2015.7348942.

[20] G. Karatas, O. Demir, and O. K. Sahingoz, "Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset," *IEEE Access*, vol. 8, pp. 32150–32162, 2020, doi: 10.1109/ACCESS.2020.2973219.

[21] K. Maharana, S. Mondal, and B. Nemade, "A review: Data pre-processing and data augmentation techniques," *Global Transitions Proceedings*, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/j.gltp.2022.04.020.

[22] N. Konstantinou and N. W. Paton, "Feedback driven improvement of data preparation pipelines," *Information Systems*, vol. 92, p. 101480, Sep. 2020, doi: 10.1016/j.is.2019.101480.

[23] A. A. A. Fernandes, M. Koehler, N. Konstantinou, P. Pankin, N. W. Paton, and R. Sakellariou, "Data Preparation: A Technological Perspective and Review," *SN Computer Science*, vol. 4, no. 4, p. 425, Jun. 2023, doi: 10.1007/s42979-023-01828-8.

[24] F. Ridzuan and W. M. N. Wan Zainon, "A Review on Data Cleansing Methods for Big Data," *Procedia Computer Science*, vol. 161, pp. 731–738, 2019, doi: 10.1016/j.procs.2019.11.177.

[25] S. K. Singh and Dr. R. K. Dwivedi, "Data Mining: Dirty Data and Data Cleaning," *SSRN Electronic Journal*, 2020, doi: 10.2139/ssrn.3610772.

[26] J. Tang, W. Chen, K. Wang, Y. Zhang, and D. Liang, "Probability-based label enhancement for multi-dimensional classification," *Information Sciences*, vol. 653, p. 119790, Jan. 2024, doi: 10.1016/j.ins.2023.119790.

[27] J. T. Hancock and T. M. Khoshgoftaar, "Survey on categorical data for neural networks," *J Big Data*, vol. 7, no. 1, p. 28, Dec. 2020, doi: 10.1186/s40537-020-00305-w.

[28] M. K. Dahouda and I. Joe, "A Deep-Learned Embedding Technique for Categorical Features Encoding," *IEEE Access*, vol. 9, pp. 114381–114391, 2021, doi: 10.1109/ACCESS.2021.3104357.

[29] S. Bagui, D. Nandi, S. Bagui, and R. J. White, "Machine Learning and Deep Learning for Phishing Email Classification using One-Hot Encoding," *Journal of Computer Science*, vol. 17, no. 7, pp. 610–623, Jul. 2021, doi: 10.3844/jcssp.2021.610.623.

[30] D. Seca and J. Mendes-Moreira, "Benchmark of Encoders of Nominal Features for Regression," in *Trends and Applications in Information Systems and Technologies*, Á. Rocha, H. Adeli, G. Dzemyda, F. Moreira, and A. M. Ramalho Correia, Eds., Cham: Springer International Publishing, 2021, pp. 146–155.

[31] C. Nkikabahizi, W. Cheruiyot, and A. Kibe, "Chaining Zscore and feature scaling methods to improve neural networks for classification," *Applied Soft Computing*, vol. 123, p. 108908, Jul. 2022, doi: 10.1016/j.asoc.2022.108908.

[32] D. Protić et al., "Numerical Feature Selection and Hyperbolic Tangent Feature Scaling in Machine Learning-Based Detection of Anomalies in the Computer Network Behavior," *Electronics*, vol. 12, no. 19, p. 4158, Oct. 2023, doi: 10.3390/electronics12194158.

[33] U. M. Khaire and R. Dhanalakshmi, "Stability of feature selection algorithm: A review," *Journal of King Saud University – Computer and Information Sciences*, vol. 34, no. 4, pp. 1060–1073, Apr. 2022, doi: 10.1016/j.jksuci.2019.06.012.

[34] S. Tangirala, "Evaluating the Impact of GINI Index and Information Gain on Classification using Decision Tree Classifier Algorithm*," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 2, 2020, doi: 10.14569/IJACSA.2020.0110277.

[35] J. Motl and P. Kordik, "Stratified Cross-Validation on Multiple Columns," in *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, Washington, DC, USA: IEEE, Nov. 2021, pp. 26–31. doi: 10.1109/ICTAI52525.2021.00012.

[36] L. Xavier and R. Thirunavukarasu, "A Distributed Tree-based Ensemble Learning Approach for Efficient Structure Prediction of Protein," *IJIES*, vol. 10, no. 3, pp. 226–234, Jun. 2017, doi: 10.22266/ijies2017.0630.25.

[37] A. A. Ceran, Y. Ar, Ö. Ö. Tanrıöver, and S. Seyrek Ceran, "Prediction of software quality with Machine Learning-Based ensemble methods," *Materials Today: Proceedings*, vol. 81, pp. 18–25, 2023, doi: 10.1016/j.matpr.2022.11.229.

[38] N. Sharma, M. Mangla, S. N. Mohanty, and C. R. Pattanaik, "Employing stacked ensemble approach for time series forecasting," *International Journal of Information Technology*, vol. 13, no. 5, pp. 2075–2080, Oct. 2021, doi: 10.1007/s41870-021-00765-0.

[39] M. Kumar, S. Singhal, S. Shekhar, B. Sharma, and G. Srivastava, "Optimized Stacking Ensemble Learning Model for Breast Cancer Detection and Classification Using Machine Learning," *Sustainability*, vol. 14, no. 21, p. 13998, Oct. 2022, doi: 10.3390/su142113998.

[40] J. Zheng, M. Wang, T. Yao, Y. Tang, and H. Liu, "Dynamic Mechanical Strength Prediction of BFRC Based on Stacking Ensemble Learning and Genetic Algorithm Optimization," *Buildings*, vol. 13, no. 5, p. 1155, Apr. 2023, doi: 10.3390/buildings13051155.

[41] A. Shahraki, M. Abbasi, and Ø. Haugen, "Boosting algorithms for network intrusion detection: A comparative evaluation of Real AdaBoost, Gentle AdaBoost and Modest AdaBoost," *Engineering Applications of Artificial Intelligence*, vol. 94, p. 103770, Sep. 2020, doi: 10.1016/j.engappai.2020.103770.

[42] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, and M. Asadpour, "Boosting methods for multi-class imbalanced data classification: an experimental

review," *J Big Data*, vol. 7, no. 1, p. 70, Dec. 2020, doi: 10.1186/s40537-020-00349-y.

[43] N. Tabassum et al., "Semantic Analysis of Urdu English Tweets Empowered by Machine Learning," *Intelligent Automation & Soft Computing*, vol. 29, no. 3, pp. 175–186, 2021, doi: 10.32604/iasc.2021.018998.

[44] D. Valero-Carreras, J. Alcaraz, and M. Landete, "Comparing two SVM models through different metrics based on the confusion matrix," *Computers & Operations Research*, vol. 152, p. 106131, Apr. 2023, doi: 10.1016/j.cor.2022.106131.

Biographies

Fauzi Adi Rafrastara, received Bachelor and Master degree in Computer Science from Universitas Dian Nuswantoro (2009) and Universiti Teknikal Malaysia Melaka (2011), respectively. He is currently serving as lecturer and researcher in Departement of Informatics Engineering, Faculty of Computer Science, Universitas Dian Nuswantoro, Indonesia. His research interest includes artificial intelligence and information security.

Guruh Fajar Shidik, received Bachelor of Computer Science from Universitas Dian Nuswantoro (UDINUS), Semarang, Indonesia in 2009, received Master of Computer Science from Technical Malaysia Melaka University

(UTeM) in 2011. He received his Doctorate degree from Universitas Gadjah Mada (UGM), Indonesia in 2016. Currently, he is a lecturer in Universitas Dian Nuswantoro (UDINUS). His area of interests includes cloud computing, wireless communication, machine learning, and artificial intelligence.

Wildanil Ghozi is a lecturer and researcher in Departement of Informatics Engineering, Faculty of Computer Science, Universitas Dian Nuswantoro, Semarang, Indonesia. He received bachelor's and master's degrees in computer science from Universitas Dian Nuswantoro in 2016 and 2019. His research interest includes artificial intelligence and information security.

Nova Rijati is with Informatics Departement, Universitas Dian Nuswantoro, Semarang, Jawa Tengah, Indonesia. She received Bachelor degree in Mathematics Departement at Universitas Diponegoro, Semarang, in 1995 and Master degree in Informatics Enginering, STTIBI, Jakarta in 2001. She is an earned doctor degree in Intelligent Electrical and Informatics Technology from Institut Teknologi Sepuluh Nopember, Surabaya, in 2021. Her research interests are computer science, artificial intelligence and data mining. She is an IAENG and IEEE member.

Oki Setiono is a lecturer and researcher at the Faculty of Health Sciences, in the Medical Records and Health Information Study Program, Universitas Dian Nuswantoro, Semarang, Indonesia. He obtained his Bachelor's and Master's degrees in Computer Science from Universitas Dian Nuswantoro in 2014 and 2017. His research interests are cloud computing.