Artificial Intelligence Evolution
http://ojs.wiserpub.com/index.php/AIE/

UNIVERSAL WISER PUBLISHER

Research Article

Enhanced Ensemble-Based Distributed Denial-of-Service (DDoS) Attack Detection with Novel Feature Selection: A Robust Cybersecurity Approach

Md. Alamgir Hossain

Department of Computer Science and Engineering, Prime University, Dhaka 1216, Bangladesh
Email: alamgir.cse14.just@gmail.com

Received: 6 July 2023; **Revised:** 12 August 2023; **Accepted:** 15 August 2023

Abstract: One of the major threats to computer networks and systems is distributed denial-of-service (DDoS) attacks. These attacks include saturating the targeted system with a large volume of traffic coming from several sources, which causes a service interruption. Detecting these attacks in real-time has become a critical task in cybersecurity. The existing method of DDoS attack detection suffers from the problem of high false positive rates. Additionally, the classifiers used in the existing methods may not be able to capture the complex patterns of the DDoS attack traffic, leading to low accuracy. In this research, I propose an enhanced approach for detecting DDoS attacks using an ensemble-based random forest classifier with a novel feature selection technique. The ability of the ensemble-based Random Forest Classifier to aggregate many decision trees to increase classification accuracy makes it a better option for DDoS attack detection than a single machine learning-based classifier. By lowering the variance and bias of the classifier, ensemble-based approaches are known to reduce overfitting and increase the robustness of the model. To choose the most useful characteristics for DDoS attack detection, the feature selection strategy uses a novel combination of correlation analysis, mutual information, and principal component analysis techniques. A part of the CIC-DDoS2019 dataset is used for the evaluation of the proposed method and to compare it to other modern approaches. The experimental results reveal that when integrated with additional evaluation metrics, the proposed approach outperforms existing techniques in various aspects, including accuracy, recall, precision, F1-score, false positive rate, and more. The proposed approach obtained almost 100% accuracy, 0% false positive rate, and 100% true positive rate.

***Keywords*:** DDoS attack detection, novel feature selection to detect DDoS attacks, ensemble-based approach to detect DDoS attacks, machine learning, cyber security

1. Introduction

A particular type of cyberattack known as a distributed denial-of-service (DDoS) attack uses a large number of compromised devices, often known as a "botnet", to attack a target server or network with a large amount of traffic, data, or requests [1-2]. A DDoS attack seeks to disturb authorized users by making the targeted system inaccessible or performing slowly. DDoS attacks can be launched from anywhere in the globe, and since they are spread, they can be challenging to stop or effectively eliminate [3-5]. They are often used by hackers or malicious actors to extort money or

Copyright ©2023 Md. Alamgir Hossain.
DOI: https://doi.org/10.37256/aie.4220233337
This is an open-access article distributed under a CC BY license
(Creative Commons Attribution 4.0 International License)
https://creativecommons.org/licenses/by/4.0/

to disrupt the operations of a business, government, or organization [6]. DDoS attacks can be highly damaging, leading to financial losses, reputation damage, and even legal consequences [7].

DDoS attacks can compromise sensitive information and put users' personal data at risk, which can have severe legal and ethical implications. These attacks are becoming increasingly common and sophisticated, making them more challenging to detect and prevent [8]. DDoS assaults may be launched by attackers using a variety of methods and technologies, and both their frequency and intensity are continuously increasing. DDoS attacks can have far-reaching consequences beyond the targeted organization. For example, a DDoS attack on a critical infrastructure provider could cause widespread disruptions, affecting the operations of other businesses, governments, and individuals. Therefore, solving the problem of DDoS attacks is not only critical for individual organizations but also for the larger community and society. Therefore, it is essential to create efficient techniques and tools for identifying and reducing DDoS attacks [9].

Due to the rising complexity of DDoS attacks, their problem-solving is challenging. DDoS attacks can target different layers of the network, making it difficult to implement effective countermeasures [10]. Furthermore, DDoS attacks can involve a large number of sources, making it difficult to distinguish legitimate traffic from attack traffic. These challenges require innovative and collaborative approaches to develop effective solutions for mitigating the impact of DDoS attacks.

We are proposing an enhanced approach for detecting DDoS attacks using an ensemble-based random forest classifier with a novel feature selection technique. For the purpose of choosing relevant features, correlation analysis, mutual information, and principal component analysis are all combined. Then from the various ensemble-based machine-learning approaches, the random forest is applied to the model. The proposed approach attempts to reduce false positives while increasing the accuracy of DDoS attack detection. We evaluate the proposed approach using a real-world dataset named CIC-DDoS2019 and show that it outperforms existing techniques in terms of accuracy, precision, recall, and other evaluation metrics. Our key contributions to this research are listed below:

- Researching the most recent DDoS attack detection techniques and evaluating their advantages and disadvantages.
- Evaluate the performance of several machine learning techniques for identifying DDoS attacks with the model.
- Developing a hybrid feature selection approach that will be effective for an ensemble-based machine learning classifier.
- Developing a machine learning framework based on ensembles that combine several classifiers to increase detection accuracy.
- Evaluating the effectiveness of the proposed ensemble-based strategy against current DDoS detection techniques.

The advantage of the ensemble-based random forest approach over the existing methods to detect DDoS attacks lies in its ability to combine the predictions of multiple decision trees to improve the accuracy of the classification. The random forest strategy can prevent overfitting and increase the robustness of the model by reducing the variance and bias of the classifier by utilizing an ensemble of classifiers rather than a single classifier. Moreover, the ensemble-based approach is better able to capture the complex patterns of DDoS attack traffic, which can lead to higher accuracy in detecting attacks compared to existing methods. Moreover, the ensemble-based approach is known for its scalability and can efficiently process large amounts of data, making it well-suited for detecting DDoS attacks in real-time scenarios.

The remaining section of this paper discusses related research for detecting DDoS attacks. In the next section, every part of the proposed model development is described. The fourth section contains the results and discussion along with essential figures and tables. The fifth section of the paper addresses the conclusion of this research.

2. Review of the literature

To identify DDoS attacks, a number of methods have been proposed over time including rule-based, statistical, machine-learning-based, etc. The most recent DDoS attack detection methods are reviewed in this section, along with their benefits and drawbacks.

In rule-based techniques, a set of rules is created to recognize DDoS attacks [11]. The parameters of the traffic flow, such as the packet rate, packet size, and protocol type, are often the basis for these rules. Although rule-based techniques are straightforward and simple to use, they might not be able to identify novel or advanced DDoS attacks that do not follow predefined rules [12]. In addition, rule-based approaches could identify genuine traffic as an attack because of

their high false positive rate. To find irregularities in network traffic, statistical approaches employ statistical models [13]. These methods examine how the traffic moves and search for changes from the typical traffic patterns. Statistical approaches are superior to rule-based solutions because they can identify both known and unidentified attacks. To understand the typical traffic patterns, statistical approaches may need a lot of training data, which might be difficult to get. Furthermore, statistical methods may have a high false alarm rate, resulting in the designation of legitimate traffic as an attack.

Machine learning techniques involve the use of algorithms to learn the patterns of normal and attack traffic and use this knowledge to detect DDoS attacks. The multi-scale base CNN technique presented by Cheng et al. [14] in 2020 to identify DDoS obtained 74% accuracy, which is quite low, and a very low True Positive Rate (TPR). The same year, Sambangi and Gondi [15] introduced a method of multiple linear regression with a 75% accuracy rate and a very high False Positive Rate (FPR).

An Intrusion Detection System (IDS) framework that integrates a group of feature engineering methods with the use of deep neural networks was suggested by Lopes et al. [16] in 2021. Nearly 99% accuracy was attained. Despite having an IDS framework, it can only identify DDoS attacks. To detect DDoS attacks, Dasari and Devarakonda suggested yet another machine learning (ML)-based model. They used different single ML-based classifiers [17]. The model with logistic regression produced the best result from the performance study, with an accuracy of 99.61%. The specificity in this case was about 84%. Since it uses a single classifier, the performance varies depending on the type of DDoS attack. SVC-RF-based classifiers with a 98.8% accuracy were proposed by Ahuja et al. Only the SDN environment is suitable for this complex model [18].

In 2022, Nuiaa et al. [19] suggested improved optimization techniques for the detection of DDoS attacks. The model with the K-Nearest Neighbor (KNN) classifier produced a result of 89.59%. Regarding their research, they recommended using additional methods like clustering or neural networks to increase the detection rate and reduce the false alarm rate. In the same year, Elgendy et al. [20] released DTEXNet, a cutting-edge method with a 95% accuracy rate. This is more difficult since it combines two neural network models. And the accuracy should be improved in relation to the dataset's size.

In 2023, Sabir [21] applied BayesNet, KNN, and J48 classifiers to detect DDoS and found the J48 classifier-based model produce the best result with an accuracy of 98.31%. But this single classifier is not workable to detect other datasets or newly patterned attacks.

With the current feature selection method, the current DDoS detection model demonstrates that not all types of DDoS detection are compatible with single-classifier-based machine learning models. Additionally, compared to the existing techniques, accuracy, and TPR should be increased, while FPR should be decreased. Although some offer much better solutions but newly patterned DDoS cannot be tackled by the existing models [22-24]. Therefore, it is necessary to develop a DDoS attack detection model that would be effective against all types of attacks. In order to provide network security against all types of DDoS attacks, the suggested technique of employing an Ensemble-based Random Forest (ERF) machine learning classifier for identifying DDoS attacks could be able to provide a more comprehensive and efficient solution.

3. Proposed model development

The proposed method for identifying DDoS attacks is thoroughly described in this section. The architecture of the pipeline for developing a machine learning-based model, including the feature selection, is shown in Figure 1.

3.1 *Dataset*

The CIC-DDoS2019 dataset is publicly available and used in this research to evaluate the model for detecting DDoS attacks. It contains network traffic data from various types of DDoS attacks, as well as legitimate traffic, and can be used to develop and test machine-learning models that find attacks involving DDoS. It contains over 16 million network flows with 88 features, making it one of the largest and most comprehensive DDoS datasets available [25]. This research exclusively utilizes LDAP-type attacks from this dataset.

3.2 *Data preprocessing*

The preprocessing steps in the model's implementation include removing duplicates, transforming infinite and large values to Not a Numbers (NaNs), deleting rows with NaNs, separating numerical and categorical columns, normalizing numerical columns, encoding categorical columns, and transforming the target variable into a discrete variable.

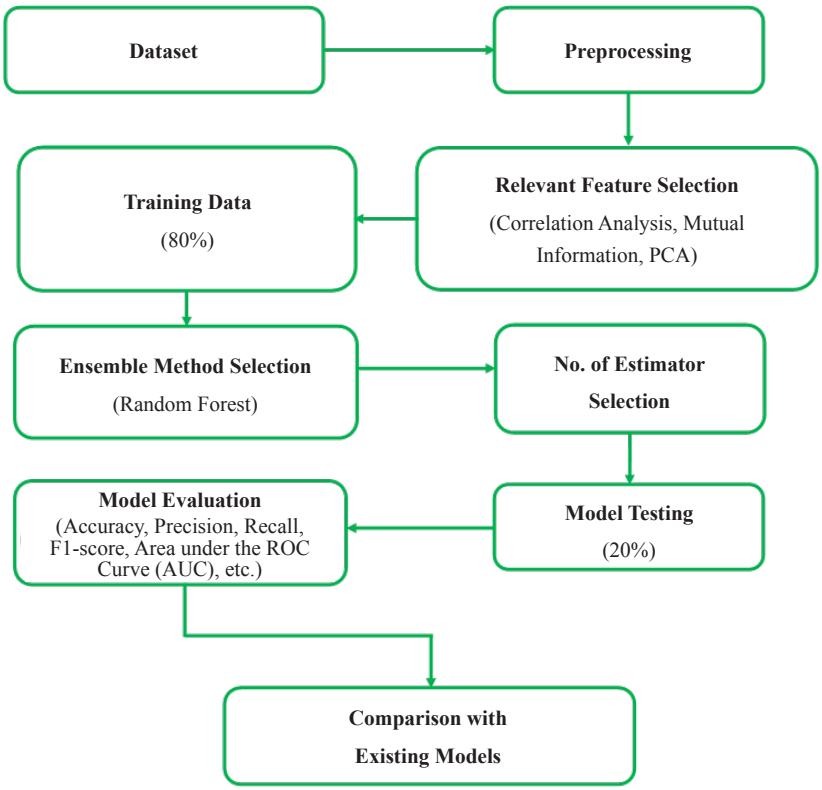


Figure 1. Proposed model-developing pipeline

3.3 *Relevant feature selection*

The relevant features in this model have been selected using correlation analysis, mutual information, and principal component analysis.

The correlation coefficient between each pair of variables in the dataset is determined via correlation analysis [26]. The correlation coefficient, whose values range from -1 (perfectly negative correlation) to 1 (perfectly positive correlation), assesses the degree and direction of the linear relationship between two variables.

The most commonly used correlation coefficient is the Pearson correlation coefficient, which is defined as:

$$r = (\Sigma(x_i - \bar{x})(y_i - \bar{y}))/sqrt(\Sigma(x_i - \bar{x})^2\Sigma(y_i - \bar{y})^2) \quad (1)$$

Where r is the Pearson correlation coefficient, x_i and y_i are the values of the two variables for the i^{th} observation, $\bar{x}$ and $\bar{y}$ are the sample means of the two variables, and Σ denotes the sum over all observations.

In this model, the relevant_features_corr list is generated by selecting the columns in the dataset that have an absolute correlation coefficient greater than 0.5.

A statistical metric known as "mutual information" estimates the amount of knowledge one variable offers about another one. It is based on information theory and can capture nonlinear relationships between variables [27]. The

mutual information between two random variables X and Y is defined as:

$$I(X; Y) = \Sigma\Sigma p(x_i, y_j) log(p(x_i, y_j)/(p(x_i)p(y_j))) \quad (2)$$

Where $I(X; Y)$ is the mutual information between X and Y, $p(x_i, y_j)$ is the joint probability mass function of X and Y, $p(x_i)$ is the probability mass function of X, and $p(y_j)$ is the probability mass function of Y. Overall conceivable X and Y values, the sum is calculated.

The mutual information estimator ranges from 0 to infinity, where 0 indicates no information gain and higher values indicate greater information gain. In feature selection, the mutual information estimator is often used to rank the variables based on their ability to discriminate between different classes or outcomes. The top-ranked variables are then selected for further analysis. The top 20 features with the highest mutual information scores are chosen for this model using the "*SelectKBest*" function from the "*sklearn.feature_selection*" package.

The method of principal component analysis (PCA) [28] is used to reduce the number of dimensions in multivariate data processing. It seeks to identify the main components-a more condensed set of uncorrelated variables-that best depict the diversity of the original dataset. In mathematics, the data matrix X, which may be written as follows, is subjected to a singular value decomposition (SVD) in order to derive the principal components.

$$X = U\Sigma V^T \quad (3)$$

Where U is a matrix of left singular vectors, Σ is a diagonal matrix of singular values, and V is a matrix of right singular vectors. The right singular vectors provide the loadings, or weights, of the original variables on each principal component, while the singular values show the percentage of variation explained by each principal component. PCA is often used for feature selection by selecting the top-ranked variables based on their loadings on the first few principal components. This may help in determining which factors are crucial for revealing the variation in the data. The PCA function from the sklearn.decomposition module is utilized in the model's implementation to conduct PCA with 20 components. The lists of features from all three techniques are then combined to create a list of the features that are the most important. Equations are below from selections to combine and select the relevant features:

$$relevant\_features\_corr = corr\_abs[corr\_abs > 0.5].index.tolist() \quad (4)$$

$$relevant\_features\_mutual = X.columns[mutual\_info.get\_support()].tolist() \quad (5)$$

$$relevant\_features\_pca = X.columns[pca.components\_.argmax(axis=1)].tolist() \quad (6)$$

$$relevant\_features = list(set().union(relevant\_features\_corr, relevant\_features\_mutual, relevant\_features\_pca)) \quad (7)$$

After employing the three mentioned techniques in this research, the relevant features are stored in "*relevant_features*". The "*relevant_features*" is a list that contains the selected relevant features obtained from three different feature selection methods: correlation analysis, mutual information, and Principal Component Analysis (PCA). Each feature selection method independently identifies a subset of features that are most relevant or informative for the detection of DDoS attacks. These methods were applied to identify features that are deemed important for improving the performance of this model. Combining these steps, the overall time complexity of the combined feature selection approach can be approximated as follows:

$$O(n^2)(Correlation\ Analysis) + O(n \times m + n \times log(n))(Mutual\ Information) + O(n^2 \times m + n \times k)(PCA)$$

3.4 *Ensemble-based random forest classifier selection*

The use of ensemble-based machine learning models for intrusion detection is promising since it increases detection rates and attack resistance [24, 29]. These models incorporate numerous separate models to increase the

prediction's overall accuracy and resilience, particularly for the detection of DDoS assaults. The diversity of the various classifiers in the ensemble allows the DDoS assaults detection system to detect a wide range of attack kinds and patterns. From the various ensemble techniques, the random forest ensemble classifier provides the best performance for the proposed model.

The Random Forest Ensemble Classifier is a powerful machine learning algorithm used for the detection of DDoS attacks in network traffic and distinguishing them from normal flows.

Let D be the training dataset containing N instances of network traffic flows, where each instance x_i is represented by a feature vector with m features: $D = \{(x\_1, y\_1), (x\_2, y\_2), ..., (x\_N, y\_N)\}$, where y_i is the corresponding class label, indicating whether the flow is a DDoS attack ($y\_i = 1$) or a normal flow ($y\_i = 0$).

Algorithm 1 illustrates the Ensemble Random Forest classifier's procedure, and Table 1 lists the hyperparameters that have been used in the model's development.

Algorithm 1: Working process of the Ensemble Random Forest classifier to Detect DDoS Attack.

1. Initialize the ensemble of decision trees T as an empty set.
2. For $t = 1$ to T:
   a. Randomly select m features from the set of input features.
   b. Create a new decision tree T_t by recursively splitting the training data D into smaller subsets based on the selected m features:
      i. At each node of the tree, choose the feature that maximizes the information gain or other splitting criterion.
      ii. Stop splitting when the maximum depth of the tree is reached or when all the instances at a node belong to the same class.
   c. Add the decision tree T_t to the ensemble T.
3. For each instance x_i in the training set D:
   a. Create a feature vector z_i by extracting the relevant features selected using techniques such as correlation analysis, mutual information, and principal component analysis (PCA).
   b. For each decision tree T_t in the ensemble T:
      i. Calculate the class prediction y_i, t using the decision path of the instance x_i in the tree T_t.
   c. Aggregate the predictions of all the decision trees in the ensemble T to obtain the final class prediction y_i:
      i. If the majority of decision trees predict $y\_i = 1$, classify x_i as a DDoS instance.
      ii. Otherwise, classify x_i as a normal instance.
4. Output the ensemble of decision trees T.

Random Forest constructs multiple decision trees, each trained on a different subset of the data and a random subset of features [29]. By combining the predictions of multiple decision trees through majority voting, the classifier is able to effectively detect DDoS attacks and distinguish them from normal flows with high accuracy and robustness. This diversity curtails overfitting, enhances generalization, and hampers the dominance of noisy features, a crucial trait when dealing with complex, high-dimensional datasets. Each tree employs a bootstrapped subset of the training data. This means different trees see different instances, introducing diversity. Mathematically, if we denote the individual trees as classifiers $h\_1, h\_2, ..., h\_n$, their collective output is akin to:

$$Ensemble\ Output = sign(h\_1(x) + h\_2(x) + ... + h\_n(x)) \tag{8}$$

Where x is the input instance. This diversity reduces variance and helps prevent overfitting.

Random Forest employs a subset of features at each split within a tree. Let's say the total features are F and $m << F$ features are randomly selected. Mathematically, if we have a split criterion Gini impurity, the probability of selecting any specific feature i is m/F. This subsampling dampens the impact of noisy features and mitigates their undue influence [30]. The process of averaging predictions from multiple trees, each with its unique biases and variances, reduces the overall variance while maintaining a reasonable bias. This striking balance aids in capturing complex relationships in the data while resisting overfitting. Random Forest utilizes the Out of Bag (OOB) samples (instances not used during a specific tree's training) for error estimation. This offers a reliable estimate of the model's generalization performance,

making it easier to tune hyperparameters [31]. It can potentially shatter the data by partitioning it into 2^d (a single decision tree with depth d) regions. By averaging multiple such trees, Random Forest's boundary is smoother and less prone to overfitting.

The Random Forest ensemble capitalizes on diverse trees, feature subsampling, and an inherent bias-variance equilibrium. Its mathematically grounded approach ensures robustness against noise and overfitting, making it an adept choice for DDoS attack detection than other classifiers where maintaining performance in real-world scenarios is paramount.

The hyperparameters in the Random Forest classifier for the proposed model are given in Table 1.

Table 1. Hyperparameters in the Random Forest classifier

Parameter	Values	Parameter	Values
n_estimators	10	min_impurity_decrease	0.0
criterion	'gini'	max_depth	None
n_jobs	None	oob_score	False
bootstrap	True	min_samples_split	2
min_samples_leaf	1	random_state	42
min_weight_fraction_leaf	0.0	verbose	0
max_features	'sqrt'	warm_start	False
max_leaf_nodes	None	class_weight	None

4. Experimental results and analysis

The Scikit-learn package and the Python programming language are used throughout the whole experiment for the implementation of the model. Google Colaboratory, commonly referred to as "Colab", is a tool developed by Google Research that have used for the experiment. Anyone can create and run arbitrary Python code using Colab, making it ideal for machine learning, data analysis, and educational applications. To be more precise, Colab is a hosted Jupyter Notebook service [32].

4.1 *Evaluation metrics*

The proportion of cases that are properly categorized to all instances defines accuracy. Accuracy indicates how effectively the model can differentiate between legitimate traffic and malicious traffic in the context of DDoS attack detection [33]. The mathematical formula for accuracy is:

$$Accuracy = (TN + TP)/(TP + TN + FP + FN) \tag{9}$$

False Positives (FP) are instances of normal traffic that were mistakenly classified as DDoS attacks, while True Positives (TP) are instances of correctly predicted DDoS attacks, True Negatives (TN) are instances of correctly predicted normal traffic, and False Negatives (FN) are instances of correctly predicted DDoS attacks.

FPR quantifies the percentage of instances among all negative instances that are mistakenly identified as positive (i.e., as a DDoS attack) [34]. A high FPR suggests that too many typical scenarios are being classified as DDoS attacks

by the model. FPR may be mathematically represented as:

$$FPR = FP/(TN + FP) \tag{10}$$

The ratio of genuine positives to the total of true positives and false positives is known as precision. Precision in the context of recognizing DDoS attacks evaluates the proportion of expected assaults that really occur as opposed to false alarms. Precision can be stated mathematically as:

$$Precision = True\ Positives/(False\ Positives + True\ Positives) \tag{11}$$

The ratio of true positives to the total of true positives and false negatives is known as recall. In the context of DDoS attack detection, recall measures how many of the actual attacks are detected by the model. Mathematically, recall can be expressed as:

$$Recall = True\ Positives/(False\ Negatives + True\ Positives) \tag{12}$$

The harmonic mean of recall and accuracy is known as the F1-score. It is a helpful indicator for assessing how accuracy and recall are traded off. The mathematical formula for the F1-score is:

$$F1\text{-}score = 2 \times ((Recall \times Precision)/(Recall + Precision)) \tag{13}$$

Cohen's Kappa is a statistical indicator of inter-rater agreement that is frequently used to assess the effectiveness of machine learning models [35]. It assesses the degree of agreement between expected and observed labels while accounting for the probability of chance agreement. Cohen's Kappa formula is as follows:

$$Kappa = (Observed\ Accuracy - Expected\ Accuracy)/(1 - Expected\ Accuracy) \tag{14}$$

Where Observed Accuracy is the percentage of properly categorized occurrences, and Expected Accuracy is the percentage of correctly identified instances that would occur by chance.

The percentage of actual attacks via DDoS that the model correctly identifies as such is measured by the TPR metric. TPR may be described mathematically as:

$$TPR = True\ Positives/(False\ Negatives + True\ Positives) \tag{15}$$

The percentage of cases that the model incorrectly categorizes is known as the error rate. It is the balancing act to accuracy [36]. The mathematical formula for the error rate is:

$$Error\ Rate = (False\ Negatives + False\ Positives)/(True\ Positives + False\ Negatives + False\ Positives + True\ Negatives) \tag{16}$$

Balanced accuracy is the average of the TPR and the true negative rate (TNR) [37]. It provides a more accurate measure of performance when the dataset is imbalanced (i.e., when there are many more instances of one class than the other). Mathematically, balanced accuracy can be expressed as:

$$Balanced\ Accuracy = (TPR + TNR)/2 \tag{17}$$

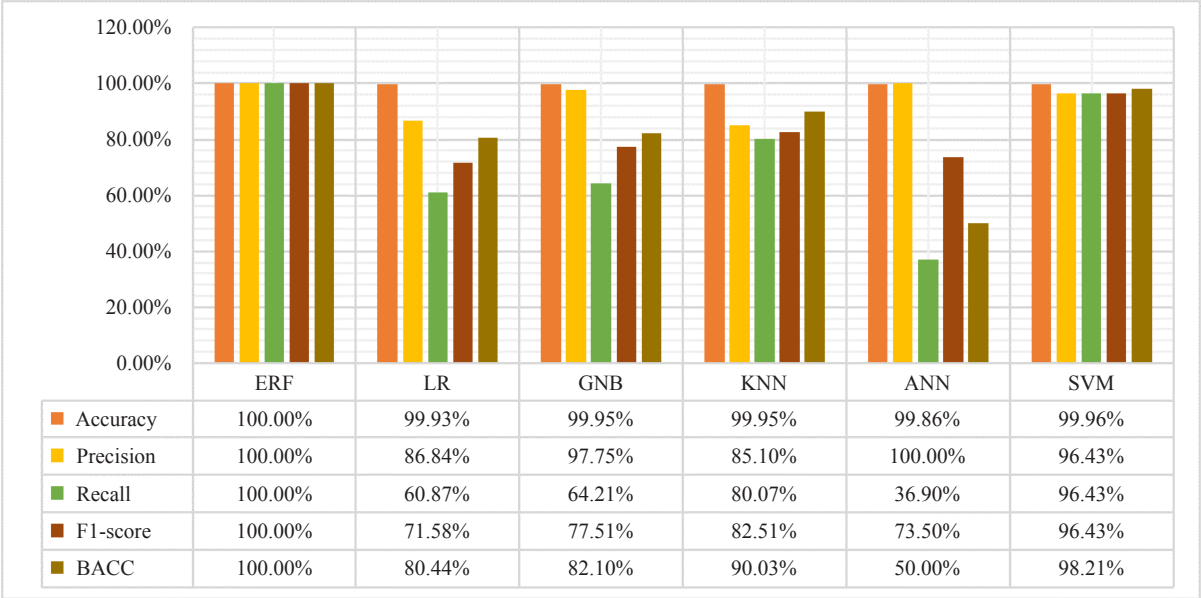
The test accuracy is the accuracy of the model on a different test dataset that it has not seen during training. The training accuracy is the accuracy of the model on the training data. The test accuracy is a more important metric as it measures the generalization performance of the model. When a model performs well on training data but badly on test

data, this is known as overfitting.

4.2 *Analysis of the findings*

The whole dataset is divided into training and testing data using the scikit-learn (sklearn) package's "*train_test_split*" function. The remaining 20% of the data is utilized for testing, while 80% is used for training. The research centers on the analysis of a specific DDoS attack dataset, focusing exclusively on LDAP attacks. Within this dataset, we have access to an extensive collection of 1,281,542 entries, encompassing a total of 87 features that pertain to this particular type of DDoS attack. After the preprocessing, the SMOTE (Synthetic Minority Over-sampling Technique) is applied. SMOTE is a data augmentation method designed to tackle class imbalance in machine learning [38]. It works by generating synthetic samples for the minority class, effectively leveling the playing field for training. By creating new instances that bridge the gap between existing ones, SMOTE enhances the model's ability to accurately classify underrepresented classes, leading to more robust and balanced predictions.

The proposed model using the ERF and five additional classical machine learning classifiers for the identification of DDoS attacks in cyber security are shown in performance metrics in Figure 2. The metrics accuracy, recall, precision, Balanced Accuracy (BACC), and F1-score have been utilized to evaluate the models in this figure. The model using the ERF classifier has achieved excellent results on all measures, demonstrating that it is a highly effective classifier for identifying attacks that use DDoS in cyber security. This means that the ERF model has correctly classified all instances without any false positives or false negatives. The accuracy score of the model with the single classifier proves that the relevant feature selection approach is also promising.



	ERF	LR	GNB	KNN	ANN	SVM
Accuracy	100.00%	99.93%	99.95%	99.95%	99.86%	99.96%
Precision	100.00%	86.84%	97.75%	85.10%	100.00%	96.43%
Recall	100.00%	60.87%	64.21%	80.07%	36.90%	96.43%
F1-score	100.00%	71.58%	77.51%	82.51%	73.50%	96.43%
BACC	100.00%	80.44%	82.10%	90.03%	50.00%	98.21%

Figure 2. Performance of the model with classical ML classifier
Note: Artificial Neural Network (ANN); Support Vector Machine (SVM)

The model's performance with several ensemble-based classifiers is shown in Table 2 in terms of recall, FPR, AUC score, and testing time in seconds. The four ensemble-based classifiers used in the evaluation are ERF, Bagging, Ada Boosting, and Simple Stacking. Based on the presented metrics, the ERF classifier outperforms the other classifiers in detecting DDoS attacks. It achieved a perfect recall score of 1.0 and an FPR of 0.0, indicating that it correctly identified all DDoS attacks without incorrectly labeling any legitimate traffic as an attack. Moreover, it obtained a high AUC score of 1.0, indicating that it has excellent overall performance in distinguishing between attack and normal traffic. However, it had the lowest testing time of 0.29684 seconds, which is relatively faster than the other classifiers. Therefore, ERF is

an effective choice of classifier in an attack with a DDoS identification model.

Table 2. Performance of the model with different ensemble-based classifiers

Classifier	Recall	FPR	AUC Score	Testing Time (Seconds)
ERF	1.00000	0.00000	1.00000	0.29684
Bagging	0.99631	0.00369	0.99815	0.89521
Ada Boosting	0.99668	0.00420	0.99623	0.36795
Gradient Boosting	0.99999	0.01570	0.99213	0.20979
Simple Stacking	0.99631	0.00369	0.99815	10.61270

Figure 3 is a heatmap of the confusion matrix, which visually represents the performance of the model in predicting the classes of the test data. The rows of the heatmap correspond to the actual values of the test data, while the columns correspond to the predicted values. Table 3 shows the values of the confusion matrix generated from the test data, including the number of true positives (TP), false negatives (FN), false positives (FP), and true negatives (TN).

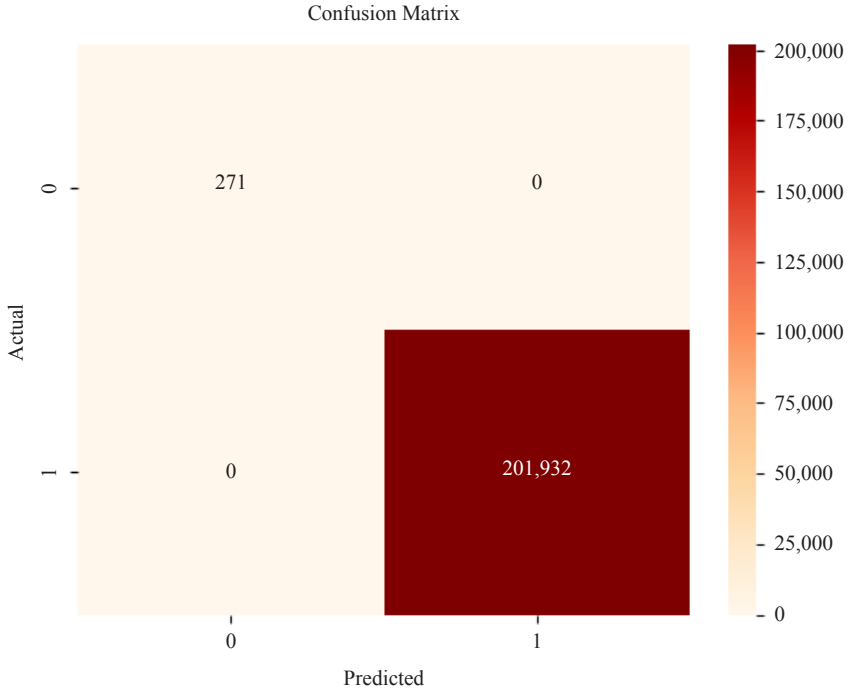


Figure 3. Confusion matrix of the test data

The model correctly classified 201,932 instances as positive (true positive) and 271 instances as negative (true negative), while there were no false negatives or false positives. This means that the model has a perfect true positive rate and false positive rate, indicating that it has performed extremely well in detecting DDoS attacks.

Table 3. Values of the confusion matrix of the test result

Test Data (20%) (202203)		Predicted Values	
		Positive	Negative
Actual Values	Positive	TP = 201,932	FN = 0
	Negative	FP = 0	TN = 271

Figure 4 shows a line plot of the training and test accuracy of the model with the ERF Classifier for different numbers of trees. The x-axis, which ranges from 1 to 100, depicts the number of trees in the forest, and the y-axis, the model's accuracy. For the training accuracy and the test accuracy, two lines are displayed. The figure demonstrates that after a given number of trees, both training accuracy and test accuracy continue to rise to 100%. This suggests that adding more trees beyond this point does not improve the model's accuracy. The optimal number of trees for this dataset appears to be around 10-20, as this is where the test accuracy is highest.

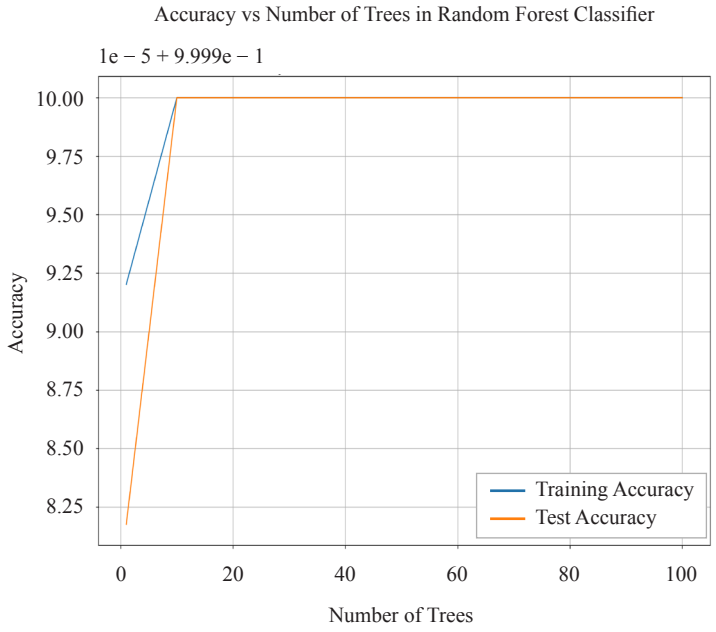


Figure 4. Accuracy of the model for different numbers of trees in the ERF classifier

The effectiveness of certain additional assessment criteria for the identification of attacks involving DDoS is shown in Table 4. The table contains the values for training accuracy, test accuracy, Cohen's Kappa, observed accuracy (Po), expected accuracy (Pe), and error rate. The model has achieved high accuracy for both the training and test sets (1.00000), indicating that the model has been able to learn the patterns of DDoS attacks very well. Cohen's Kappa is also high (1.00000), indicating that the model's predictions are in excellent agreement with the actual values. The observed accuracy (Po) is 1.00000, which is the proportion of correct predictions. The expected accuracy (Pe) is 0.99732, which is the expected proportion of correct predictions if the predictions were made randomly. The error rate is 0.00000, indicating that the model did not make any errors in the classification of the test data. Overall, the model has performed very well in the detection of DDoS attacks. Overall, these metrics suggest that the model is highly accurate and effective

for the detection of DDoS attacks.

Table 4. Performance of some other evaluation metrics

Training Accuracy	Test Accuracy	Cohen's Kappa		Error Rate
		Observed Accuracy (Po)	Expected Accuracy (Pe)	
1.00000	1.00000	1.00000	0.99732	0.00000

The ROC curve displayed in Figure 5, is the performance of a binary classifier for detecting DDoS attacks, where the positive class represents the DDoS attacks and the negative class represents non-attack traffic. The FPR is shown on the x-axis, while the TPR is shown on the y-axis, also known as sensitivity or recall. The curve shows how well the classifier is able to distinguish between the two classes at different probability thresholds. The value of the ROC curve (AUC = 1.00000) indicates that the model has a perfect classification performance and can distinguish between positive and negative classes with 100% accuracy.

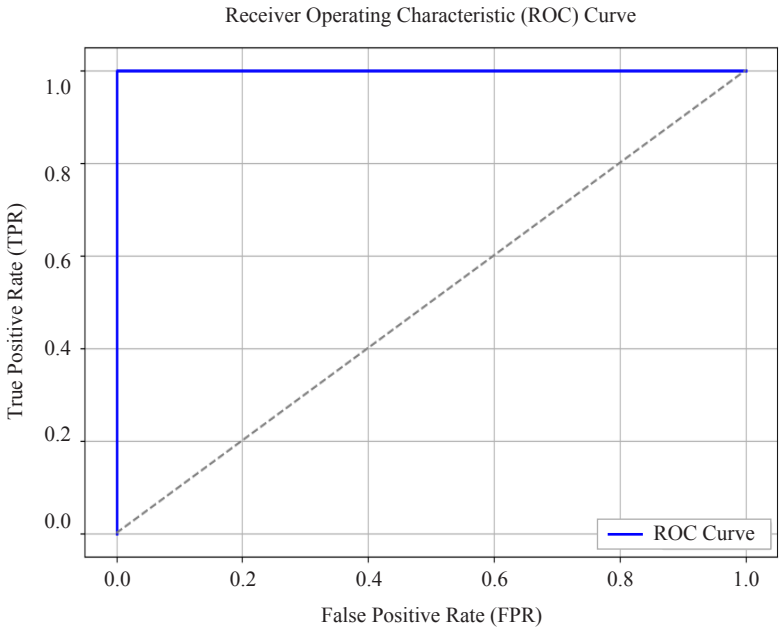


Figure 5. ROC curve of TPR vs FPR

The Precision-Recall curve, a key evaluation tool for evaluating the effectiveness of a binary classification model, is shown in Figure 6. The graph shows the dynamic interaction between recall (the proportion of actual positive instances correctly identified by the model) and precision (the ability of the model to accurately identify positive instances) at different probability thresholds. According to this graph, the model successfully detects positive instances while limiting false positives, resulting in high precision and recall at the same time.

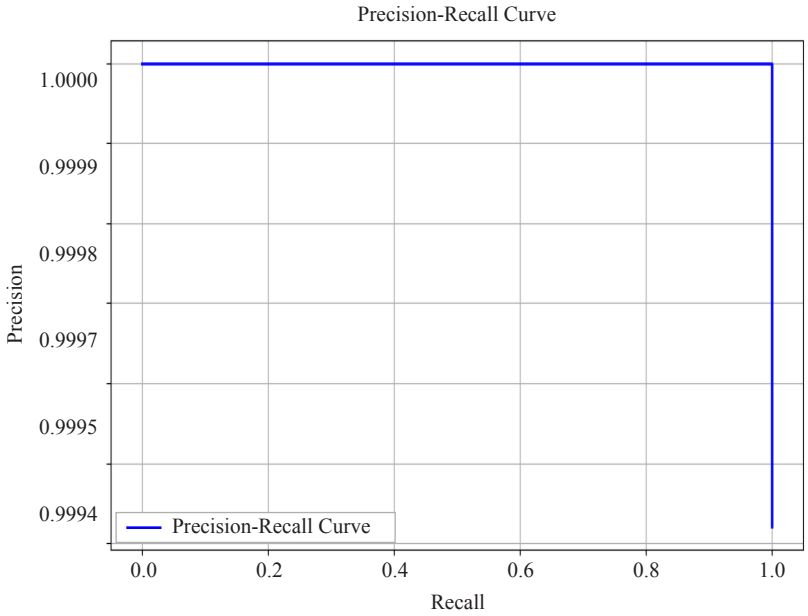


Figure 6. Precision-Recall curve

The ERF classifier-based approach employed for identifying the presence of attacks using DDoS in Figure 7 shows the log-loss values against the predicted probability for each class. The plot shows two lines, one for each class (Class 0 and Class 1), and each line represents the log-loss values for that class as the predicted probabilities range from their minimum to their maximum values. The projected probability is represented on the x-axis, while the log-loss value is represented on the y-axis. The log-loss metric measures the performance of a classification model by penalizing false classifications. A lower log-loss value indicates better performance, with a perfect classifier having a log-loss of 0. In the figure generated from the code, the blue line represents the log-loss for class 1 (DDoS attack) and shows that the log-loss value remains constant at 0 for all predicted probabilities. This suggests that the model performs perfectly in predicting class 1, with no false positives or false negatives.

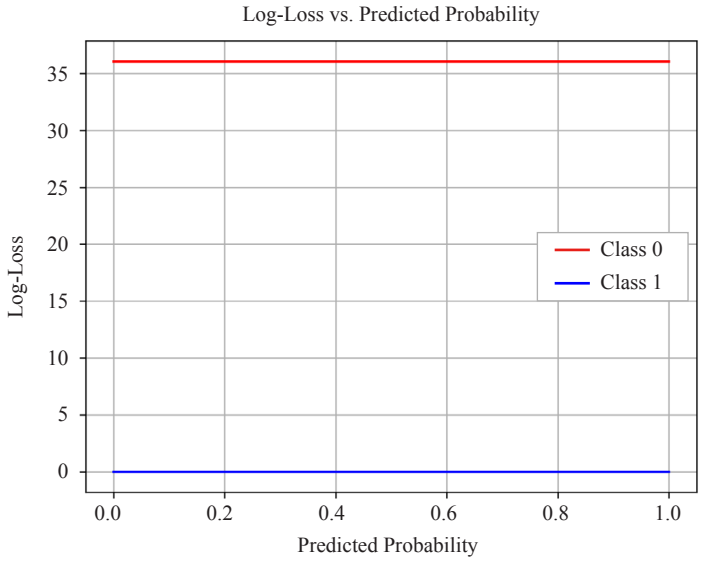


Figure 7. Predicted probabilities for DDoS attack and normal class data

Figure 8 underscores the model's robustness and effectiveness in DDoS attack detection under varying degrees of noise or variations. While the metrics experience a gradual decline as noise increases, the model's ability to maintain relatively higher Precision suggests its potential to accurately identify positive instances, which is crucial in the context of detecting DDoS attacks. The figure provides valuable insights into the model's behavior in noisy scenarios, aiding in understanding its real-world performance and guiding potential improvements.

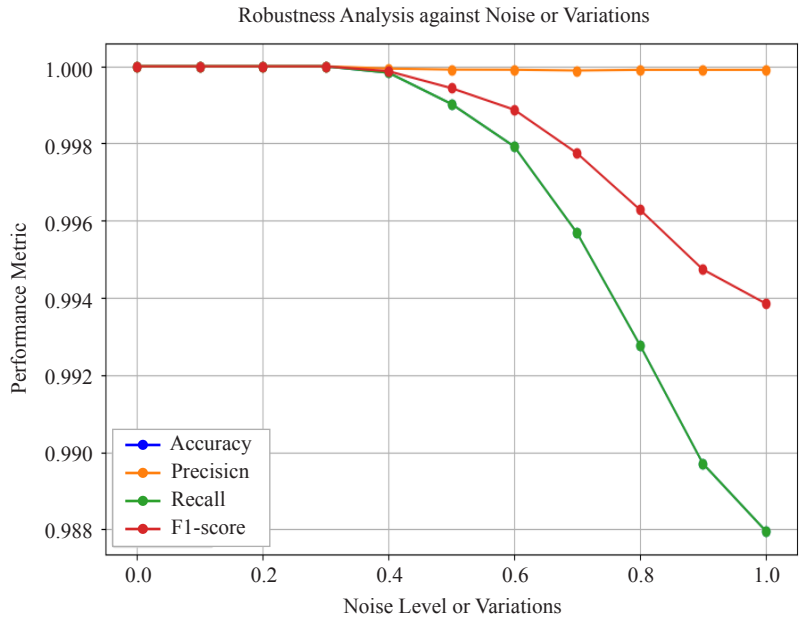


Figure 8. Robustness of the model

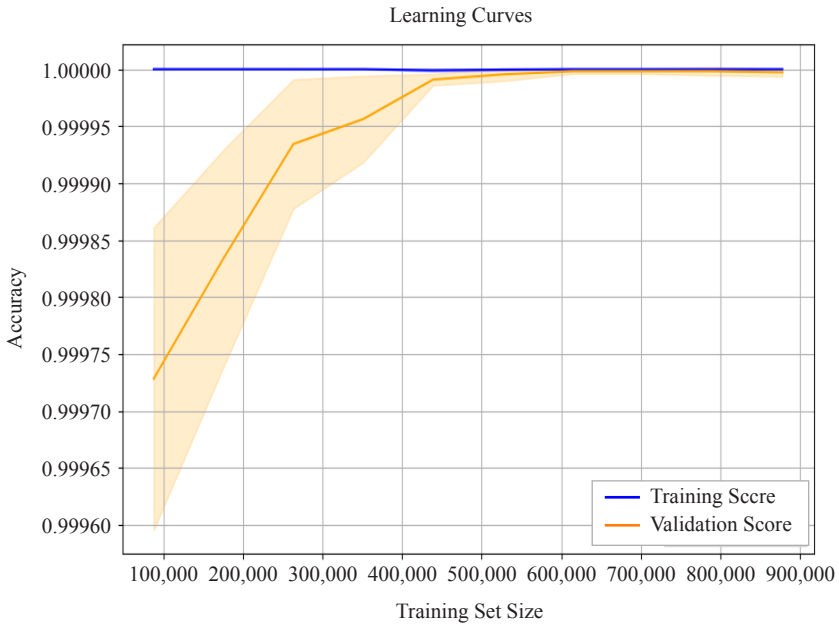


Figure 9. The learning curve of the model

The Learning Curve, shown in Figure 9, is used to evaluate how well the model performs as the size of the training dataset changes. It displays two separate curves that when compared to various training set sizes, show the training score (in blue) and the validation score (in orange). As the training set size grows, both curves converge and are stable at high accuracy levels, which suggests a well-generalized and reliable model. This result indicates that the model successfully learns from the data and performs consistently on hypothetical cases, indicating its capacity to identify underlying patterns without overfitting.

4.3 *Comparing the performance with the existing models*

Table 5 compares the effectiveness of several DDoS attack detection models using various assessment metrics for the CIC-DDoS2019 Dataset. Outperforming all other currently available techniques, the suggested technique achieves an accuracy of 100%. The proposed model outperforms the other current models, indicating that it is significantly more successful in terms of cyber security. This shows that the suggested methodology is quite accurate and effective in identifying attacks with DDoS. Moreover, the ERF classifier helps the model to generalize better by reducing the variance and increasing the robustness of the model. It also enables the model to handle missing data and outliers in the input dataset by considering multiple decision trees rather than depending just on one decision tree.

Table 5. Evaluation of the demonstrated model's performance in comparison with currently available models

Model Name	Accuracy (%)	Recall (%)	Precision (%)	F1-score (%)
Proposed Method	100.00	100.00	100.00	100.00
J48, 2023 [21]	98.31	98.30	98.30	98.30
DTEXNet, 2022 [20]	95.00	95.00	95.00	95.00
PFS and KNN, 2022 [19]	89.59	90.04	89.64	89.84
CyDDoS, 2021 [16]	99.60	99.70	99.70	99.60
LR, 2021 [17]	99.66	99.76	99.89	99.83
SVC-RF, 2021 [18]	98.80	97.91	98.27	97.65
CNN and BILSTM, 2021 [39]	94.52	92.04	94.74	93.44
FF-DL, 2021[40]	79.00	79.00	78.00	78.00
MLR, 2020 [15]	75.00	75.00	75.00	75.00
MS-CNN, 2020 [14]	74.00	74.00	74.00	74.00

Overall, the presented model for detecting DDoS attacks demonstrates outstanding performance across all evaluation metrics. Moreover, when compared to existing datasets, it consistently outperforms them. This exceptional performance positions the model as a formidable contender in the field of cybersecurity for DDoS attack detection. Its accuracy, precision, recall, and F1-score indicate its reliability and effectiveness in identifying and mitigating DDoS threats. As a result, this model holds great promise in bolstering cyber defense strategies and safeguarding against malicious attacks. Its robustness and superior results make it a valuable asset for enhancing the security landscape in the ever-evolving digital realm.

5. Conclusion and future direction

The proposed model has showcased remarkable performance across various assessment parameters, displaying promising and effective results in the domain of DDoS attack detection. The exemplary AUC score of 1 underscores the model's exceptional discriminatory power in distinguishing between normal network traffic and DDoS attacks, substantiating its robustness and reliability. By achieving a 100% accuracy rate, the demonstrated methodology holds tremendous potential to significantly bolster the recognition and response capabilities for countering DDoS attacks, thereby fortifying the overall cyber security posture of organizations and networks. Moreover, the innovative utilization of the ERF-based model addresses the limitations of single classifier-based approaches, enabling the detection of novel and previously unknown types of security threats, and extending the model's versatility beyond DDoS attacks. With its superior performance and adaptability, the proposed approach can be seamlessly integrated into real-time systems such as network firewalls, intrusion detection systems, and network traffic monitoring tools, culminating in a comprehensive defense mechanism against diverse cyber threats.

In future research, the focus could be directed toward optimizing the model's performance in scenarios where network traffic is significantly higher and rapidly changing. This may involve investigating techniques for parallelization, distributed computing, or hardware acceleration to ensure the model's effectiveness in handling substantial data volumes. Additionally, exploring ways to integrate the ERF-based model into advanced threat detection systems that operate in cloud environments or utilize Software-Defined Networking (SDN) architectures could be another fruitful direction. Investigating how the model can leverage cloud-based resources and adapt to dynamic network conditions would be essential for enhancing its scalability and responsiveness.

Conflict of interest

The author declares that he has no conflicts of interest that could potentially influence or bias the findings and conclusions presented in this research.

References

[1] Khare M, Oak R. Real-time Distributed Denial-of-Service (DDoS) attack detection using decision trees for server performance maintenance. In: Pant M, Sharma TK, Basterrech S, Banerjee C. (eds.) *Performance Management of Integrated Systems and its Applications in Software Engineering*. Singapore: Springer Singapore; 2020. p.1-9. Available from: http://link.springer.com/10.1007/978-981-13-8253-6_1 [Accessed 30th Apr 2023].

[2] Kumari P, Jain AK. A comprehensive study of DDoS attacks over IoT network and their countermeasures. *Computers & Security*. 2023; 127: 103096.

[3] Cheema A, Tariq M, Hafiz A, Khan MM, Ahmad F, Anwar M. Prevention techniques against distributed denial of service attacks in heterogeneous networks: A systematic review. *Security and Communication Networks*. 2022; 2022: 1-15.

[4] De Neira AB, Kantarci B, Nogueira M. Distributed denial of service attack prediction: Challenges, open issues and opportunities. *Computer Networks*. 2023; 222: 109553.

[5] Mishra A, Gupta N, Gupta BB. Defensive mechanism against DDoS attack based on feature selection and multi-classifier algorithms. *Telecommunication Systems*. 2023; 82(2): 229-244.

[6] Alawida M, Omolara AE, Abiodun OI, Al-Rajab M. A deeper look into cybersecurity issues in the wake of Covid-19: A survey. *Journal of King Saud University-Computer and Information Sciences*. 2022; 34(10): 8176-8206.

[7] Bao AC. What is Ddos in Cyber Security. 2023. Available from: https://www.alibabacloud.com/topic-center/cyber-security/ghcxip49kf-what-is-ddos-in-cyber-security#:~:text=For%20businesses%2C%20DDoS%20attacks%20can,or%20even%20cause%20physical%20damage [Accessed 2th Feb 2023].

[8] Sujatha G, Kanchhal Y, George G. An advanced approach for detection of Distributed Denial of Service (DDoS) attacks using machine learning techniques. *2022 3rd International Conference on Smart Electronics and Communication (ICOSEC)*. Trichy, India: IEEE; 2022. p.821-827. Available from: https://ieeexplore.ieee.org/

document/9951944/.
[9] Azure Network Security Team. *2022 in review: DDoS attack trends and insights*. Microsoft Security. 2023. Available from: https://www.microsoft.com/en-us/security/blog/2023/02/21/2022-in-review-ddos-attack-trends-and-insights/ [Accessed 3th May 2023].
[10] Eliyan LF, Di Pietro R. DoS and DDoS attacks in software defined networks: A survey of existing solutions and research challenges. *Future Generation Computer Systems*. 2021; 122: 149-171.
[11] Prriyadarshini MA, Renuka Devi S. Detection of DDoS attacks using supervised learning technique. *Journal of Physics: Conference Series*. 2020; 1716(1): 012057.
[12] Abu Bakar R, Huang X, Javed MS, Hussain S, Majeed MF. An intelligent agent-based detection system for DDoS attacks using automatic feature extraction and selection. *Sensors*. 2023; 23(6): 3333.
[13] Simmross-Wattenberg F, Asensio-Perez JI, Casaseca-de-la-Higuera P, Martin-Fernandez M, Dimitriadis IA, Alberola-Lopez C. Anomaly detection in network traffic based on statistical inference and \alpha-stable modeling. *IEEE Transactions on Dependable and Secure Computing*. 2011; 8(4): 494-509. Available from: https://doi.org/10.1109/TDSC.2011.14.
[14] Cheng J, Liu Y, Tang X, Sheng VS, Li M, Li J. DDoS attack detection via multi-scale convolutional neural network. *Computers, Materials & Continua*. 2020; 62(3): 1317-1333. Available from: https://doi.org/10.32604/cmc.2020.06177.
[15] Sambangi S, Gondi L. A machine learning approach for DDoS (Distributed Denial of Service) attack detection using multiple linear regression. *The 14th International Conference on Interdisciplinarity in Engineering-INTER-ENG 2020*. Târgu Mureş, Romania: MDPI; 2020. p.51. Available from: https://www.mdpi.com/2504-3900/63/1/51.
[16] Ortet Lopes I, Zou D, Ruambo FA, Akbar S, Yuan B. Towards effective detection of recent DDoS attacks: A deep learning approach. *Security and Communication Networks*. 2021; 2021: 1-14.
[17] Dasari KB, Devarakonda N. Detection of different DDoS attacks using machine learning classification algorithms. *Information System Engineering*. 2021; 26(5): 461-468. Available from: https://doi.org/0.18280/isi.260505.
[18] Ahuja N, Singal G, Mukhopadhyay D, Kumar N. Automated DDOS attack detection in software defined networking. *Journal of Network and Computer Applications*. 2021; 187: 103108.
[19] Nuiaa RR, Manickam S, Alsaeedi AH, Alomari ES. A new proactive feature selection model based on the enhanced optimization algorithms to detect DRDoS attacks. *International Journal of Electrical and Computer Engineering*. 2022; 12(2): 1869.
[20] Elgendy S, Attawiya M, Haridy O, Farag A, Branco P. DTEXNet: Artificial intelligence-based combination scheme for DDoS attacks detection. *The 35th Canadian Conference on Artificial Intelligence.* Torronto, Canada: Canadian Artificial Intelligence Association; 2022.
[21] Sabir MY. *DDoS Attacks Detection using Machine Learning*. Visakhapatnam, India: Gandhi Institute of Technology and Management; 2023. Available from: http://hdl.handle.net/1828/15046 [Accessed 29th Apr 2023].
[22] Barni M, Campisi P, Delp EJ, Doërr G, Fridrich J, Memon N, et al. Information forensics and security: A quarter-century-long journey. *IEEE Signal Process Magezine*. 2023; 40(5): 67-79.
[23] Abu Al-Haija Q, Alohaly M, Odeh A. A lightweight double-stage scheme to identify malicious DNS over HTTPS traffic using a hybrid learning approach. *Sensors*. 2023; 23(7): 3489.
[24] Hossain MA, Islam MS. Ensuring network security with a robust intrusion detection system using ensemble-based machine learning. *Array*. 2023; 19: 100306. Available from: https://doi.org/10.1016/j.array.2023.100306.
[25] Sharafaldin I, Lashkari AH, Hakak S, Ghorbani AA. Developing realistic Distributed Denial of Service (DDoS) attack dataset and taxonomy. *2019 International Carnahan Conference on Security Technology (ICCST)*. CHENNAI, India: IEEE; 2019. p.1-8. Available from: https://ieeexplore.ieee.org/document/8888419/.
[26] Duangsoithong R, Windeatt T. Correlation-based and causal feature selection analysis for ensemble classifiers. *Artificial Neural Networks in Pattern Recognition: 4th IAPR TC3 Workshop (ANNPR 2010)*. Cairo, Egypt: Springer Berlin Heidelberg; 2010. p.25-36. Available from: http://link.springer.com/10.1007/978-3-642-12159-3_3.
[27] Macedo F, Valadas R, Carrasquinha E, Oliveira MR, Pacheco A. Feature selection using decomposed mutual information maximization. *Neurocomputing*. 2022; 513: 215-232.
[28] Odhiambo OE, Onyango OG, Waema KM. Feature selection for classification using principal component analysis and information gain. *Expert Systems with Applications*. 2021; 174: 114765.
[29] Chauhan NS. *Random Forest®-A Powerful Ensemble Learning Algorithm*. KDnuggets; 2020. Available from: https://www.kdnuggets.com/2020/01/random-forest-powerful-ensemble-learning-algorithm.html [Accessed 27th Feb 2023].
[30] Palczewska A, Palczewski J, Marchese Robinson R, Neagu D. Interpreting random forest classification models using a feature contribution method. In: Bouabana-Tebibel T, Rubin SH. (eds.) *Integration of Reusable Systems.*

Advances in Intelligent Systems and Computing. Cham: Springer International Publishing; 2014. p.193-218. Available from: https://link.springer.com/10.1007/978-3-319-04717-1_9.
[31] Chan JCW, Paelinckx D. Evaluation of random forest and adaboost tree-based ensemble classification and spectral band selection for ecotope mapping using airborne hyperspectral imagery. *Remote Sensing of Environment*. 2008; 112(6): 2999-3011.
[32] Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, et al. Scikit-learn: Machine learning in Python. *Journal of machine Learning Research*. 2011; 12: 2825-2830.
[33] Vujovic ŽĐ. Classification model evaluation metrics. *International Journal of Advanced Computer Science and Applications*. 2021; 12(6): 599-606.
[34] Agrawal A, Singh R, Khari M, Vimal S, Lim S. Autoencoder for design of mitigation model for DDOS attacks via M-DBNN. *Wireless Communications and Mobile Computing*. 2022; 2022: 1-14.
[35] Vieira SM, Kaymak U, Sousa JMC. Cohen's kappa coefficient as a performance measure for feature selection. *International Conference on Fuzzy Systems*. Barcelona, Spain: IEEE; 2010. p.1-8. Available from: http://ieeexplore.ieee.org/document/5584447/.
[36] Grandini M, Bagli E, Visani G. *Metrics for Multi-Class Classification: An Overview*. arXiv [Preprint]. 2020. Available from: http://arxiv.org/abs/2008.05756.
[37] Chicco D, Jurman G. A statistical comparison between Matthews correlation coefficient (MCC), prevalence threshold, and Fowlkes-Mallows index. *Journal of Biomedical Informatics*. 2023; 144: 104426.
[38] Joloudari JH, Marefat A, Nematollahi MA, Oyelere SS, Hussain S. Effective class-imbalance learning based on SMOTE and convolutional neural networks. *Applied Sciences*. 2023; 13(6): 4006.
[39] Alghazzawi D, Bamasag O, Ullah H, Asghar MZ. Efficient detection of DDoS attacks using a hybrid deep learning model with improved feature selection. *Applied Sciences*. 2021; 11(24): 11634.
[40] Cil AE, Yildiz K, Buldu A. Detection of DDoS attacks with feed forward based deep neural network model. *Expert Systems with Applications*. 2021; 169: 114520.

Appendix

Feature of the dataset with description

Feature Name	Description of the Feature
Idle Min	Represents the minimum time duration in seconds that a flow remains idle, indicating periods of inactivity between packets.
Flow IAT Max	Denotes the maximum inter-arrival time between two consecutive packets of a flow in seconds, revealing the longest time gap between packets.
Active Min	Signifies the minimum time duration in seconds that a flow remains active, reflecting the shortest duration of flow activity.
FIN Flag Count	Indicates the number of packets with the FIN (Finish) flag set in the flow, which is a common flag used in closing connections.
Flow Packets/s	Represents the rate of packets per second in the flow, indicating the intensity of packet transmission.
Idle Std	Represents the standard deviation of idle times in seconds for the flow, providing a measure of variation in flow inactivity periods.
Fwd Avg Bytes/Bulk	Denotes the average number of bytes per bulk transaction in the forward direction, giving insights into data transfer patterns.
Bwd IAT Min	Reflects the minimum inter-arrival time between two consecutive packets in the backward direction, indicating the shortest time gap between packets in that direction.
URG Flag Count	Signifies the number of packets with the URG (Urgent) flag set in the flow, indicating the presence of urgent data.
Flow Bytes/s	Represents the rate of bytes per second in the flow, providing an overall measure of data transfer speed.
Init_Win_bytes_backward	Denotes the window size in bytes of the last packet in the backward direction, indicating the window size for flow termination.
Source IP	Refers to the IP address of the source (sender) of the flow, identifying the origin of the network traffic.
Flow Duration	Represents the total duration in seconds of the flow, providing a measure of the flow's lifetime.
Bwd Header Length	Denotes the total length of headers in bytes in the backward direction, indicating the header overhead.
Fwd IAT Std	Represents the standard deviation of inter-arrival times in seconds in the forward direction, providing insights into packet arrival patterns.
Packet Length Std	Signifies the standard deviation of packet lengths in bytes in the flow, indicating the variability in packet sizes.
Bwd IAT Total	Denotes the total inter-arrival time in seconds for the backward direction, providing the cumulative time between packets.
Avg Bwd Segment Size	Represents the average segment size in bytes in the backward direction, indicating the average data segment size in that direction.
Inbound	A binary feature indicating whether the flow is inbound or not (1: inbound, 0: not inbound), revealing the flow's direction.
RST Flag Count	Indicates the number of packets with the RST (Reset) flag set in the flow, signifying connection reset occurrences.
Fwd Packet Length Min	Reflects the minimum packet length in bytes in the forward direction, indicating the smallest packet size.
Destination IP	Refers to the IP address of the destination (receiver) of the flow, identifying the destination of the network traffic.
Total Backward Packets	Denotes the total number of packets in the backward direction, providing an overall measure of packet count in that direction.
PSH Flag Count	Indicates the number of packets with the PSH (Push) flag set in the flow, signifying immediate data delivery.
Fwd IAT Min	Represents the minimum inter-arrival time between two consecutive packets in the forward direction, indicating the shortest time gap between packets in that direction.
act_data_pkt_fwd	Denotes the number of packets with actual data in the forward direction, excluding control packets.
Bwd Avg Bulk Rate	Represents the average bulk rate in the backward direction, indicating the rate of bulk data transmission.

Feature Name	Description of the Feature
Average Packet Size	Signifies the average size of packets in bytes in the flow, providing a measure of the typical packet size.
Down/Up Ratio	Represents the ratio of download to upload traffic in the flow, indicating the flow's traffic distribution.
Active Std	Signifies the standard deviation of active times in seconds for the flow, providing a measure of variation in flow activity durations.
Protocol	Represents the protocol used in the flow (e.g., TCP, UDP), indicating the communication protocol used.
Fwd URG Flags	Indicates the number of packets with the URG (Urgent) flag set in the forward direction, signifying urgent data delivery in that direction.
Idle Mean	Represents the mean of idle times in seconds for the flow, providing the average duration of flow inactivity periods.
Fwd Packet Length Mean	Signifies the mean packet length in bytes in the forward direction, providing the average packet size in that direction.
Bwd Packets/s	Represents the rate of packets per second in the backward direction, indicating the intensity of packet transmission in that direction.
Subflow Fwd Bytes	Denotes the total number of bytes in the forward subflow, providing the total data size in that subflow.
Total Fwd Packets	Represents the total number of packets in the forward direction, providing an overall measure of packet count in that direction.
Bwd Packet Length Mean	Signifies the mean packet length in bytes in the backward direction, providing the average packet size in that direction.
Bwd IAT Std	Represents the standard deviation of inter-arrival times in seconds in the backward direction, providing insights into packet arrival patterns in that direction.
Bwd Packet Length Std	Signifies the standard deviation of packet lengths in bytes in the backward direction, indicating the variability in packet sizes in that direction.
Fwd Avg Bulk Rate	Represents the average bulk rate in the forward direction, indicating the rate of bulk data transmission in that direction.
Fwd PSH Flags	Indicates the number of packets with the PSH (Push) flag set in the forward direction, signifying immediate data delivery in that direction.
Flow IAT Mean	Represents the mean inter-arrival time between two consecutive packets in the flow, providing the average time gap between packets.
Bwd PSH Flags	Indicates the number of packets with the PSH (Push) flag set in the backward direction, signifying immediate data delivery in that direction.
Bwd IAT Max	Denotes the maximum inter-arrival time between two consecutive packets in the backward direction, indicating the longest time gap between packets in that direction.
SimillarHTTP	A binary feature indicating whether the flow is similar to HTTP or not (1: similar, 0: not similar), providing insights into flow characteristics.
Fwd IAT Max	Represents the maximum inter-arrival time between two consecutive packets in the forward direction, indicating the longest time gap between packets in that direction.
Idle Max	Denotes the maximum time duration in seconds that a flow remains idle, indicating the longest period of inactivity between packets.
Total Length of Fwd Packets	Represents the total length of packets in bytes in the forward direction, providing an overall measure of data size in that direction.
Active Mean	Signifies the mean of active times in seconds for the flow, providing the average duration of flow activity.
Total Length of Bwd Packets	Represents the total length of packets in bytes in the backward direction, providing an overall measure of data size in that direction.
Flow IAT Std	Represents the standard deviation of inter-arrival times in seconds for the flow, indicating the variability in time gaps between packets.
Subflow Fwd Packets	Denotes the total number of packets in the forward subflow, providing an overall measure of packet count in that subflow.
Active Max	Denotes the maximum time duration in seconds that a flow remains active, indicating the longest duration of flow activity.
Destination Port	Refers to the port number of the destination (receiver) in the flow, identifying the destination port used in the communication.

Feature Name	Description of the Feature
Fwd Packet Length Max	Represents the maximum packet length in bytes in the forward direction, indicating the largest packet size in that direction.
Source Port	Refers to the port number of the source (sender) in the flow, identifying the source port used in the communication.
Timestamp	Represents the timestamp of the flow, providing the time at which the flow was observed.
Init_Win_bytes_forward	Denotes the initial window size in bytes of the first packet in the forward direction, indicating the initial data transmission window.
Flow ID	Provides a unique identifier for the flow, enabling individual flow identification.
Max Packet Length	Represents the maximum packet length in bytes in the flow, indicating the largest packet size in the entire flow.
SYN Flag Count	Indicates the number of packets with the SYN (Synchronize) flag set in the flow, signifying connection establishment requests.
Bwd Packet Length Max	Denotes the maximum packet length in bytes in the backward direction, indicating the largest packet size in that direction.
Fwd Packets/s	Represents the rate of packets per second in the forward direction, indicating the intensity of packet transmission in that direction.
Bwd Avg Bytes/Bulk	Represents the average number of bytes per bulk transaction in the backward direction, providing insights into data transfer patterns in that direction
Subflow Bwd Packets	Denotes the total number of packets in the backward subflow, which can help identify irregular packet patterns associated with certain DDoS attacks.
Avg Fwd Segment Size	Represents the average size of forward segments in bytes, providing insights into the typical segment size in the forward direction.
Bwd URG Flags	Indicates the number of packets with the URG (Urgent) flag set in the backward direction, which may indicate malicious attempts to bypass security mechanisms.
Fwd Header Length	Denotes the total length of headers in bytes in the forward direction, which can help detect anomalies or excessive header data used in some DDoS attacks.
Fwd IAT Mean	Represents the mean inter-arrival time between two consecutive packets in the forward direction, providing the average time gap between packets in that direction.
Bwd Packet Length Min	Reflects the minimum length of a packet in bytes in the backward direction, providing insights into the smallest packet size received by the destination.
Packet Length Mean	Signifies the mean length of packets in bytes in the flow, which can help establish the typical packet size and identify deviations during an attack.
min_seg_size_forward	Denotes the minimum segment size allowed in the forward direction, which can be exploited in attacks attempting to overwhelm the target with tiny segments.
Fwd Packet Length Std	Represents the standard deviation of packet lengths in bytes in the forward direction, which can help detect variations in packet sizes during an attack.
ACK Flag Count	Indicates the number of packets with the ACK (Acknowledgment) flag set in the flow, which is commonly present in legitimate TCP traffic but can also be abused in DDoS attacks.
CWE Flag Count	Indicates the number of packets with the CWE (Common Weakness Enumeration) flag set in the flow, which may suggest potential weaknesses that attackers can exploit.
Flow IAT Min	Represents the minimum inter-arrival time between two consecutive packets in the flow, providing insights into the shortest time gap between packets.
Bwd Avg Packets/Bulk	Denotes the average number of packets per bulk transaction in the backward direction, providing insights into patterns of bulk data transfer.
Bwd IAT Mean	Represents the mean inter-arrival time between two consecutive packets in the backward direction, providing the average time gap between packets in that direction.
Min Packet Length	Reflects the minimum length of a packet in bytes in the flow, which can help identify unusually small packets associated with certain attacks.
Fwd Avg Packets/Bulk	Represents the average number of packets per bulk transaction in the forward direction, providing insights into patterns of bulk data transfer.
Subflow Bwd Bytes	Denotes the total number of bytes in the backward subflow, which can help identify irregular data patterns associated with certain DDoS attacks.

Feature Name	Description of the Feature
ECE Flag Count	Indicates the number of packets with the ECE (Explicit Congestion Notification Echo) flag set in the flow, which may indicate potential network congestion due to an attack.
Packet Length Variance	Represents the variance of packet lengths in bytes in the flow, providing a measure of the spread of packet sizes, which can help identify anomalous patterns.
Fwd IAT Total	Denotes the total inter-arrival time in seconds for the forward direction, providing the cumulative time between packets in that direction.
Fwd Header Length	Represents the total length of headers in bytes in the forward direction, which can help detect anomalies or excessive header data used in some DDoS attacks.