SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Enhancing Multi-Class DDoS Attack Classification using Machine Learning Techniques

Mohammad Jawad Kadhim Abood[1,*], Ghassan Hameed Abdul-Majeed[2]

[1] Department of Information Networks, College of Information Technology, University of Babylon, Babylon, Iraq
[2] Department of Computer Engineering, College of Engineering, University of Baghdad, Baghdad, Iraq

ARTICLE INFO

Article history:
Received 8 August 2023
Received in revised form 4 October 2023
Accepted 1 March 2024
Available online 17 April 2024

Keywords:

DDoS detection; CICDDoS2019; J48; Machine learning; Classification

ABSTRACT

Distributed Denial of Service (DDoS) attacks, which adversely impact network availability, confidentiality, and integrity, represent a persistent threat. These attacks involve affected systems consuming resources through spurious requests instead of serving legitimate clients. Various methodologies exist for detecting and mitigating DDoS attacks, with Machine Learning (ML) emerging as a particularly effective approach due to its predictive capabilities after training on pertinent data. The primary objective of this study is to identify an improved ML algorithm for the detection of multiple DDoS types, considering metrics such as accuracy, precision, recall, and training time. Leveraging WEKA tools and the CICDDoS2019 dataset, several machine-learning algorithms, including Multilayer Perceptron, Reduced Error Pruning (REP) Tree, Partial Decision Tree (PART), RandomForest, and J48, were trained and evaluated. Among these, J48 was determined to be the superior algorithm for classifying four DDoS types (UDP, SYN, Portmap, MSSQL), based on the aforementioned criteria. The algorithms were experimented with using diverse sets of features, and optimal results were obtained using six features, resulting in an overall accuracy of 99.97%. Subsequently, the selected algorithm was integrated into a real-time model, exhibiting exceptional performance, which will be thoroughly elucidated and discussed in a forthcoming paper.

1. Introduction

Distributed Denial of Service (DDoS) attacks pose significant threats to network security, utilizing multiple compromised systems to overwhelm a single target system with traffic, leading to its crash. These attacks encompass various methods, such as inundating the target system with an excessive number of requests or overwhelming it with substantial data volumes. Traditional DDoS detection methods have proven time-consuming and susceptible to false positives, prompting the development of machine learning algorithms to enhance detection accuracy.

Machine learning algorithms demonstrate considerable effectiveness in detecting DDoS attacks by analysing extensive data and identifying distinctive patterns indicative of an attack. Their capacity

* *Corresponding author.*
E-mail address: mu4su@uobabylon.edu.iq

https://doi.org/10.37934/araset.43.2.7592

to swiftly and accurately detect attacks, analyse intricate patterns in large datasets, and adapt to evolving threats positions them as ideal tools for DDoS detection [1]. The primary aim of employing classification algorithms in DDoS detection systems is to effectively differentiate and categorize DDoS attack requests amidst regular traffic. In this context, the key objectives are achieving high prediction accuracy and minimizing model training durations[2].

Various factors influence accuracy and training time, with the dataset size playing a pivotal role in both. The Canadian Institute for Cybersecurity (CIC) offers meticulously curated datasets that encompass different types of DoS and DDoS attacks, aiding in research and analysis [6]. Researchers frequently explore feature selection techniques to optimize model training times by reducing dataset size and identifying pertinent features [3-5]. Additionally, configuring machine learning algorithm parameters significantly impacts performance and training efficiency, with a trade-off between accuracy and training duration often observed.

This research endeavours to compare the performance of multiple algorithm models using the CICDDoS2019 dataset [6], assessing testing time, accuracy, precision, and recall of key algorithms such as Multilayer Perceptron, Reduced Error Pruning (REP) Tree, Partial Decision Tree (PART), Random Forest, and J48. The primary objective of this project is to construct an efficient DDoS detection classifier by implementing feature selection strategies that streamline the model, resulting in reduced training time, improved accuracy, and optimal resource utilization for real-time DDoS attack detection.

2. Literature

Numerous studies have investigated the risk of DDoS attacks using a variety of methodologies. The majority of studies have employed machine learning techniques, such as classification, clustering, and prediction. The dynamic K-NN algorithm[7], ARIMA [8], SVD [9], entropy variations [10], PCA [11], MLP [12], and Naive Bayes (NB) [13] algorithms are examples of machine learning techniques.

The study [14], addresses the subject of improving the accuracy of DDoS attack detection using the well-known CICDDoS2019 dataset. In addition, the DDoS dataset was pre-processed using two primary techniques to extract the most pertinent information. Four distinct machine-learning models were chosen for use with the DDoS dataset. Based on the outcomes of real-world testing, the Random Forest machine learning model provided the highest detection accuracy, surpassing the most recently produced DDoS detection systems.

The authors of [15] presented a hybrid solution based on machine learning to detect DDoS attacks. Combining the Extreme Learning Machine (ELM) algorithm and the black-hole optimization technique constitutes the proposed system. To evaluate the efficacy of the proposed hybrid machine learning system, the authors conducted several experiments using diverse data sets. Using the CICDDoS2019 dataset, the suggested hybrid system detects DDoS attacks in cloud computing with a detection accuracy of 99.8 percent.

In contrast, the authors of [16] developed a DDoSNet Intrusion Detection System for SDN environments. The suggested approach is based on Deep Learning (DL) and integrates Recurrent Neural Networks (RNN) with auto encoders. Using the CICDDoS2019 dataset, the developed system has been assessed. The authors demonstrated a substantial improvement in attack detection over existing approaches. Thus, the proposed method provides high confidence in the protection of SDN environments.

In [17], the author examines the effect of a data balancing algorithm on the network traffic classification problem on various forms of DDoS attacks using the CICDDoS2019 dataset, which contains reflection-based and exploitation-based attack information. Results demonstrated the

efficacy of data balancing strategies such as synthetic minority sampling, naive random, and adaptive synthetic sampling in classifying network threats.

Based on multiple classification algorithms using the CICDDoS 2019 dataset, the authors of [18] proposed a detection system capable of detecting different forms of DDoS attacks. In addition, the authors recorded packets from the SDK environment, pre-processed the dataset, and then applied a classification system to detect DDoS attempts. Compared to SVM and Nave Bayes machine learning models, the authors found that the decision tree gives superior performance.

The research given in [19] analyses the success rate of the intrusion detection system using multiple machine learning techniques. Many machines learning models, including the ANN, Support Vector Machine (SVM), Gaussian Nave Bayes, Multinomial Nave Bayes, Logistic Regression, K-nearest neighbour (KNN), Decision Tree, and Random Forest algorithms, were studied using the CICDDoS2019 dataset. The authors demonstrated that K-nearest neighbour, logistic regression, and Nave Bayes provide the most accurate predictions.

Deep Neural Network (DNN) was used by the authors of [20] as a deep learning technique to detect DDoS attacks on a network traffic packet sample. Since it comprises feature extraction and classification methods, the DNN model may perform quickly and with high detection accuracy even with tiny sample sizes. The authors conducted their trials using the CICDDoS2019 dataset, which contains numerous 2019-created DDoS attack types. Using the deep learning model, the suggested system achieves an accuracy of 94.57%.

Finally, in [21], the study introduces a DDoS detection system design using a transfer learning model. Within the realm of cloud computing, securing the cloud for the Internet of Things (IoT) is a critical research focus. The devised transfer learning model integrates a 1D CNN and a decision tree model. The 1D CNN is responsible for feature extraction from the input data, and these features are subsequently utilized by the transfer learning model for classification purposes. The training phase achieved an accuracy of 97 percent, while the testing phase demonstrated a commendable accuracy of 96.33 percent.

3. Methodology

In this section, an overview of the relevant theories, concepts, principles, models and preprocessing that form the foundation for the study is presented to help to establish the context, rationale, and theoretical framework for the research.

3.1 DDoS Attack Types

There are three main types of DDoS attacks:

i. Volume-based attacks: This is the most common type of DDoS attack. It involves flooding the target system with large volumes of traffic from multiple sources. This can overload the target system and cause it to crash or become unavailable.
ii. Protocol-based attacks: This type of attack targets the protocols used by the target system. It can involve sending malformed packets or packets with a high rate of errors. This can overload the target system and cause it to crash or become unavailable.
iii. Application-based attacks: This type of attack targets the applications used by the target system. It can involve sending malicious packets or requests that exploit vulnerabilities in the target application. This can overload the target system and cause it to crash or become unavailable.

3.2 Machine Learning Algorithms

Machine learning is a rapidly growing field of Artificial Intelligence (AI) that focuses on the development of computer programs that can learn and adapt through experience. It is a powerful tool used to analyse data and make predictions based on the patterns and trends it finds. There are many different algorithms used in machine learning This research implements 5 Machine learning algorithms: J48, Multilayer perceptron (MLP), REPTree, PART, and Random Forest.

i. J48 is a type of decision tree algorithm used to create classifiers from a given dataset. It is an extension of the ID3 algorithm, which was developed by Ross Quinlan in the early 1980s. J48 works by splitting the dataset into smaller subsets using an attribute-value pair, then using the information gained to decide which attribute to split on next. This process is repeated until all subsets are pure, meaning they have only one class within them. Once the tree is built, it can be used to make classifications based on the data it has seen.
ii. Multilayer perceptron (MLP) is a type of artificial neural network that is composed of multiple layers of neurons connected. It is used for supervised learning tasks such as classification and regression. MLP networks are composed of an input layer, one or more hidden layers, and an output layer. Each layer is made up of neurons that have weighted connections to the neurons in the layers before and after it. The neurons in the hidden layers use an activation function to process the inputs and generate them.
iii. REPTree (Reduced Error Pruning Tree) is a decision tree algorithm that uses the pruning method to reduce the size of the tree. It works by recursively splitting the dataset using the attribute-value pair that maximizes the information gain. Then, it prunes the tree by removing branches that do not improve the accuracy of the tree. REPTree is often used for large datasets, as it can quickly generate a tree with good accuracy and low complexity.
iv. PART (Pruned Association Rule Tree) is an extension of the Apriori algorithm, which is used to identify frequent item sets in a dataset. PART builds a decision tree based on the frequent item sets that have been identified by the Apriori algorithm. The resulting tree can then be used to classify.
v. Finally, Random Forest is a machine learning algorithm that is used for both classification and regression problems. This algorithm works by constructing multiple decision trees from a given set of training data and then combining the predictions of the decision trees to make predictions about the data. Random Forest is often used for applications such as credit scoring and fraud detection, as it can make accurate predictions with large datasets.

3.3 Description of Dataset

The CICDoS2019 dataset is a comprehensive dataset that has been developed by the Canadian Institute for Cybersecurity (CIC) [6]. It consists of millions of records of both legitimate and malicious traffic.

The data was collected from the Canadian Institutes of Cybersecurity (CIC) honeypot network during the period from August to October 2019. The data set is designed to enable the development of new machine learning algorithms that can accurately detect and classify DDoS attacks. The data set is divided into two parts, namely the benign traffic and the malicious traffic. The benign traffic consists of normal user-initiated activities such as web browsing, email, file transfers, etc. The malicious traffic includes various types of network-based attacks such as TCP SYN floods, UDP floods, ICMP floods, etc see Table 1. All the traffic is labelled according to its type. For example, all the

malicious traffic is labelled as 'malicious' while all the benign traffic is labelled as 'benign'. The CICDDoS2019 dataset also contains several features that can be used to characterize the traffic. These features include the source and destination IP addresses, ports, packet length, payload size, protocol, flags, etc. All the records in the dataset also contain the timestamp of when the traffic was generated. This helps in understanding the timing of the attack and can be used for further analysis. The CICDDoS2019 dataset also contains a large number of attack types and patterns. Overall, the CICDDoS2019 dataset provides an excellent platform for researchers and practitioners to develop new machine-learning algorithms for accurately detecting and classifying DDoS attacks. It is a great resource for the development of new machine-learning algorithms that can accurately detect and classify DoS attacks.

Table 1
CICDDoS-2019 dataset summary

Dataset DDoS Attack Files	Label	Quantity	Ratio Percentage	The Total Number
LDAP	BENIGN	1602	0.07	2,181,530
	DDoS_LDAP	2,179,928	99.93	
MSSQL	BENIGN	1995	0.04	4,524,484
	DDoS_MSSQL	4,522,489	99.96	
DNS	BENIGN	3380	0.07	5,074,382
	DDoS_DNS	5,071,002	99.93	
NetBIOS	BENIGN	1705	0.04	4,094,978
	DDoS_NetBIOS	4,093,273	99.96	
NTP	BENIGN	14,337	1.18	1,216,976
	DDoS_NTP	1,202,639	98.82	
UDP	BENIGN	2151	0.07	3,136,794
	DDoS_UDP	3,134,643	99.93	
SNMP	BENIGN	1502	0.03	5,161,365
	DDoS_SNMP	5,159,863	99.97	
SSDP	BENIGN	762	0.03	2,611,372
	DDoS_SSDP	2,610,610	99.97	
SYN	BENIGN	389	0.03	1,380,404
	DDoS _Syn	1,380,015	99.97	

3.4 WEKA

WEKA is a collection of machine learning algorithms and tools for data mining tasks. It stands for "Waikato Environment for Knowledge Analysis" and is named after the Waikato region in New Zealand where the University of Waikato, the institution behind Weka, is located. Weka is open-source software and has become one of the most widely used platforms for machine learning and data mining research and applications. It provides a wide range of machine learning algorithms for classification, regression, clustering, association rule mining, and more. These algorithms are ready to use and can be applied to various types of data.

WEKA offers tools for data pre-processing, including cleaning, transforming, filtering, and handling missing values. This is crucial for preparing the data before feeding it to machine learning algorithms. It also includes visualization tools that allow users to explore and visualize data to better understand its characteristics and patterns. Visualization is essential for gaining insights and making informed decisions during the data analysis process.

By WEKA, Users can evaluate the performance of different algorithms, compare results, and fine-tune models based on evaluation metrics. In addition to that, there are user-friendly graphical

interface, making it accessible to both beginners and experienced data scientists. It allows users to build, train, and evaluate machine learning models without needing to write extensive code [22].

3.5 The Preprocessing

The initial download of the first CICDDoS2019 dataset included all its features. To ensure data cleanliness, any instances of Not a Number (NaN) values and duplicate columns were removed. Within this process, a redundancy was identified in the "Fwd Header Length" feature, leading to the elimination of one of them. The dataset was then narrowed down to utilize only 320,000 records from four different files. Among these records, there were 64,000 instances for each type of DDoS attack (UDP, SYN, Portmap, MSSQL) and an additional 64,000 benign records.

It is important to note that special attention was paid to addressing flawed data within the dataset. For instance, records with negative values were excluded from the dataset. Furthermore, all records with a source port or destination port value of zero were also eliminated.

To enhance the precision of outcomes, feature selection was carried out based on widely recognized criteria. A visual representation of a subset of the CICDDoS2019 dataset can be found in Figure 1.

	A	B	C	D	E	F
1	Flow ID	Source IP	Source Por	Destination	Destination	Protocol
2	192.168.10.5	104.16.207.165	443	192.168.10.5	54865	6
3	192.168.10.5	104.16.28.216	80	192.168.10.5	55054	6
4	192.168.10.5	104.16.28.216	80	192.168.10.5	55055	6
5	192.168.10.1	104.17.241.25	443	192.168.10.1	46236	6
6	192.168.10.5	104.19.196.102	443	192.168.10.5	54863	6
7	192.168.10.5	104.20.10.120	443	192.168.10.5	54871	6
8	192.168.10.5	104.20.10.120	443	192.168.10.5	54925	6
9	192.168.10.5	104.20.10.120	443	192.168.10.5	54925	6
10	192.168.10.8	104.28.13.116	443	192.168.10.8	9282	6
11	192.168.10.5	104.97.123.193	443	192.168.10.5	55153	6
12	192.168.10.5	104.97.125.160	443	192.168.10.5	55143	6
13	192.168.10.5	104.97.125.160	443	192.168.10.5	55144	6
14	192.168.10.5	104.97.125.160	443	192.168.10.5	55145	6
15	192.168.10.5	104.97.139.37	443	192.168.10.5	55254	6
16	192.168.10.1	104.97.140.32	80	192.168.10.1	36206	6
17	192.168.10.2	121.29.54.141	443	192.168.10.2	53524	6
18	192.168.10.2	121.29.54.141	443	192.168.10.2	53524	6
19	192.168.10.2	121.29.54.141	443	192.168.10.2	53526	6
20	192.168.10.2	121.29.54.141	443	192.168.10.2	53526	6
21	192.168.10.2	121.29.54.141	443	192.168.10.2	53527	6
22	192.168.10.2	121.29.54.141	443	192.168.10.2	53528	6
23	192.168.10.2	121.29.54.141	443	192.168.10.2	53527	6
24	138.201.37.2	138.201.37.241	443	192.168.10.5	55035	6
25	144.76.121.1	144.76.121.178	443	192.168.10.5	55275	6
26	145.243.233	145.243.233.16:	443	192.168.10.5	55277	6

Fig. 1. Sample of CICDDoS2019 dataset

3.6 The Feature Selection Methods

Feature selection methods play an important role in improving model performance, there is a trade-off between the number of features and the time complexity of an algorithm, the more features model leads to the most time complexity and the most accuracy, and vice versa. The number of features in the data set is 84. CICDDoS2019 is rich in features as compared to the rest of the datasets with a small number of features like the UCAL dataset [23]. Therefore, it is important to use feature selection techniques to reduce the number of features to enhance the current study.

The features were chosen using a wrapper feature selection method and information gain to reduce the number of features by ranking the features depending on the required number of features. At first, the wrapper feature selection selects 12 features Table 2, then depending of the information gain score 3 different groups selected from 10 groups.

Table 2
Features selected using the Wrapper Method

#	Feature selected
1.	Average Packet Size
2.	Destination Port
3.	Flow Bytes/s
4.	Flow Packets/s
5.	Init_Win_bytes_backward
6.	Init_Win_bytes_forward
7.	Packet Length Mean
8.	Packet Length Std
9.	Protocol
10.	Source IP
11.	Source Port
12.	Total Length of Fwd Packets

The three selected groups contain 8, 6 and 4 features as illustrated in Table 3 but all the features belong to the main wrapper group Table 2. In addition, a fourth Set is added by replacing the Flow Bytes/s feature in Set 2 with source IP feature to test the effect of the source IP in detecting the attack.

Table 3
Features Sets used in the experiment

#	Set No.	Features	Features No.
1	Set 1	Source Port, Destination Port, Protocol, Flow Bytes/s, Flow Packets/s, Packet Length Mean, Packet Length Variance, Average Packet Size	8
2	Set 2	Source Port, Destination Port, Flow Bytes/s, Flow Packets/s, Packet Length Mean, Packet Length Std	6
3	Set 3	Source Port, Destination Port, Flow Packets/s, Average Packet Size	4
4	Set 4	Source IP, Source Port, Destination Port, Flow Packets/s, Packet Length Mean, Packet Length Std	6

The ML model dataflow is depicted in Figure 2. Algorithm 1 and 2 is used to select relevant features and eliminate worthless or not applicable features.

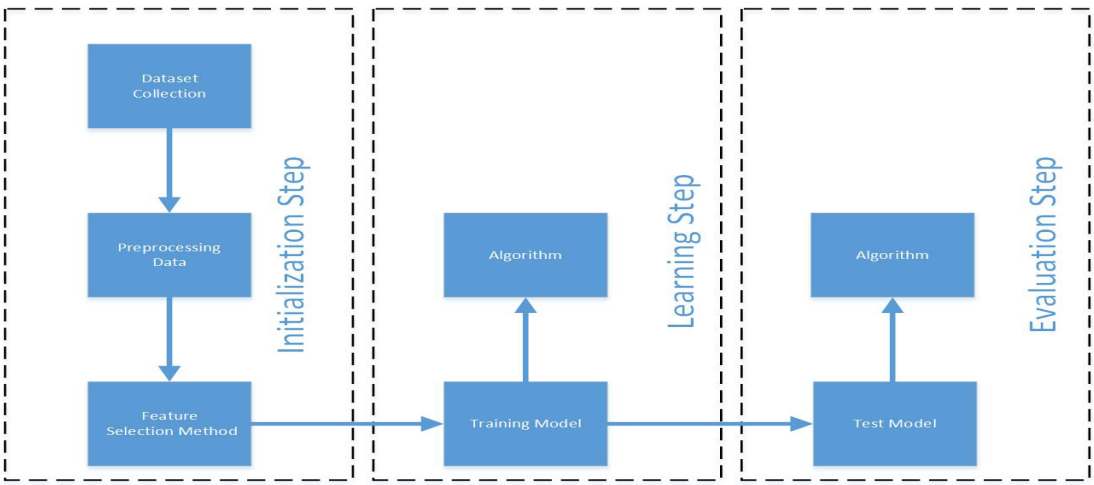


Fig. 2. The ML and features selection process

Algorithms 1 and 2 are utilized for selecting the most appropriate features that distinguish irregular traffic from normal traffic. The stage of selecting and refining features is of utmost importance within the realm of machine learning. Not all features within a dataset are employed by machine learning algorithms. Employing the complete set of features for constructing a predictive model is not only resource-intensive, but it also prolongs the process. Initially, data must undergo normalization, deduplication, or rectification to address imbalanced data. In practical scenarios, datasets such as DDoS attack traffic tend to be extensive. Therefore, to effectively train these vast datasets containing numerous features, it is imperative to eliminate irrelevant or redundant information from the dataset. Moreover, insignificant or redundant features may negatively impact the performance of the detection mechanism. The distribution of attack classes in the CICDDoS2019 dataset tends to be uneven. Additionally, the datasets encompass attributes that lack relevance when it comes to identifying attacks.

Consequently, an adopted approach for feature selection is employed to identify extraneous features among the initial set of 84 features, serving as the initial phase within the machine learning process.

Algorithm 1: Wrapper algorithm

Input: F = Training dataset, processing n features f1, f2, f3... fn
Output: The selected features list

1. Initialize an empty set S to hold the selected features.
2. Initialize a set F with all the features in the dataset.
3. While F is not empty, do the following:
3.1. For each feature f in F, add f to S, and build a J48 decision tree using S as the feature set.
3.1.1. Evaluate the performance of the decision tree.
3.1.2. Keep track of the performance for each feature and decision tree.
3.1.3. Remove the feature that resulted in the highest performance from S, if S is not empty.
3.1.4. If removing the feature decreased performance or if S is empty, stop and return the set S as the selected features.
3.2 End for
4. End while
5. End

Algorithm 2: Filter Feature Selection

Input: F: Dataset with features (X) and target variable (y).
Output: A selected subset of features

1. Calculate Feature Scores:
a. For each feature in X:
i. Apply a statistical test or compute a correlation score between the feature and the target variable.
ii. Assign a score to each feature based on the test or correlation value.
2. Rank Features:
a. Sort features in descending order based on their scores.
3. Select Top Features:
a. Choose the top-k features with the highest scores.
4. Return Selected Features.
5. End

3.7 The Classifier Selection

Utilizing the dataset, we develop classification models employing diverse machine learning techniques following the identification of optimal feature subsets through algorithms 1 and 2. Leveraging multiple learning models such as MP, REPT, PART, RF, and J48 - widely recognized supervised learning algorithms - we gauge their performance.

The outcome of the approach outlined in algorithm 3.3 yields the most effective classifier, which had been used in the study.

Algorithm 3: ClassifierSelect

Input: FL = Features_List
Output: Features Subsets, Accuracy, and Testing time with Fast and Accurate Model

```
1.  Begin
2.   For every feature Fr in Feature_Ranked data
3.    Start to Select from Feature Sets
4.     SET1/Groups1 features
5.     SET2/Groups2 features
6.     SET3/Groups3 features
7.     SET4/Groups4 features
8.      For each Feature in SETs/Groups
9.      Feed Selected features to MP, REPT, PART, RF, J48, Stacked_Modle
10.       Apply Classifier
11.          C1 = MultilayerPerseptron model
12.          C2 = REP Tree model
13.          C3 = PART model
14.          C4 = Random Forest model
15.          C5 = J48 model
16.       Calculate Test time, Tree size, and Accuracy
17.       Compare the Accuracy and testing time of C1, C2, C3, C4, and C5
18. END Algorithm
```

The evaluation criteria included both the accuracy of results and the speed of the training phase. The dataset comprised a diverse set of network packets, including both normal and malicious traffic, to provide a realistic representation of real-world network environments. The accuracy of results was assessed by measuring the algorithm's ability to correctly classify network traffic as normal or malicious. A high true positive rate and a low false positive rate were desirable to minimize both missed detections (false negatives) and false alarms (false positives). Furthermore, the training phase's speed was evaluated to ensure that the chosen algorithm could efficiently process large-scale network traffic data without excessive computational overhead. After careful analysis and comparison of the experimental results, it was found that the J48 algorithm, demonstrated outstanding performance. It achieved the highest accuracy in classifying network traffic with a true positive rate and a false positive rate. Moreover, the training phase of the model was significantly faster compared to other complex algorithms.

3.8 Performance Evaluation

To evaluate the performance of the implemented classifiers for DDoS detection, there are several metrics introduced by [24] which are Accuracy, precision, recall and performance training time has been suggested.

The four used metrics are explained shortly as the following:

i. Training time: refers to the overall amount of time necessary to train the machine learning algorithm.
ii. Accuracy: refers to the ratio of accurately predicted transactions to the total number of transactions.
Accuracy = (TP+TN)/(TP+TN+FP+FN)
iii. Precision: refers to the total predicted true transactions divided by the total predicted true transactions and false transactions.
Precision = TP/(TP+FP)
iv. Recall: refers to the total predicted number of transactions divided by the total number of actual transactions.
Recall = TP/(TP+FN)

4. Experimental Results and Discussion

This section presents the results of the experimental model developed using machine learning algorithms such as J48, MultilayerPerseptron, PART, Random Forest, and RepTree.

Weka is utilized to retrieve experimental results. As mentioned in section (Performance Evaluation) three metrics are used to measure the performance of the algorithm (training time, Accuracy, and error rate). The experiment test modes split data into 60% training and 40% testing.

The results are presented and discussed regarding evaluation metrics:

i. Training time: Table 4 illustrates that the highest training time belongs to MultilayerPerseptron (MP) and random forest (RF) algorithms respectively when using set 1 (Figure 3) which contains 8 features. In contrast, the lowest time registered to REPTree and J48 respectively when using Set 3 which contains 4 features (Figure 5). Figure 4 and Figure 6 show the estimated time to train the classifier using Set 2 and Set 4

Table 4
Illustrates the training time for 5 algorithms using 4 sets of features

Algorithm	Training Time in Seconds			
	Set 1	Set 2	Set 3	Set 4
J48	3.7	2.8	2.23	2.54
MP	148.8	118.14	88.03	88.89
REPTree	2.38	1.44	0.96	1.01
PART	7.15	5.74	4.28	4.84
RF	68.86	56.14	58.57	45.44

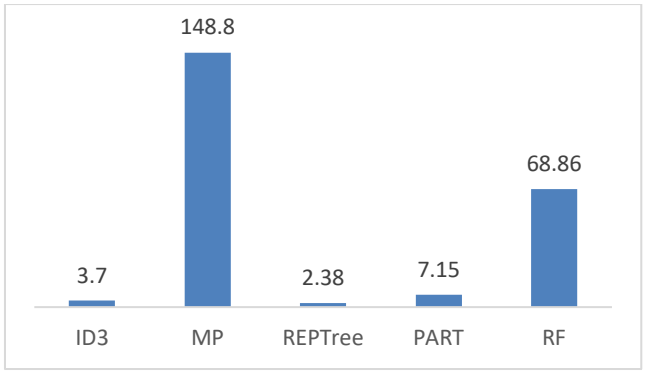


Fig. 3. Training Time Using Set 1

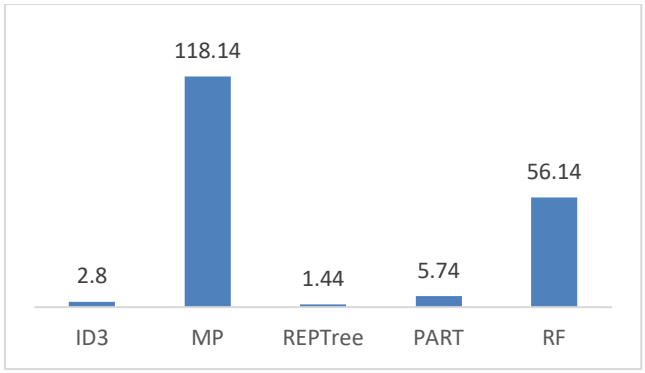


Fig. 4. Training Time Using Set 2

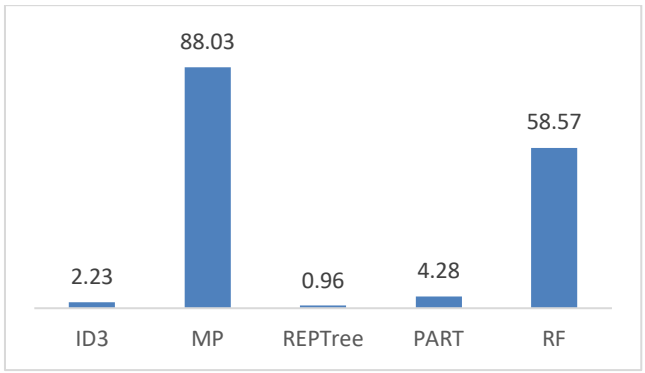


Fig. 5. Training Time Using Set 3

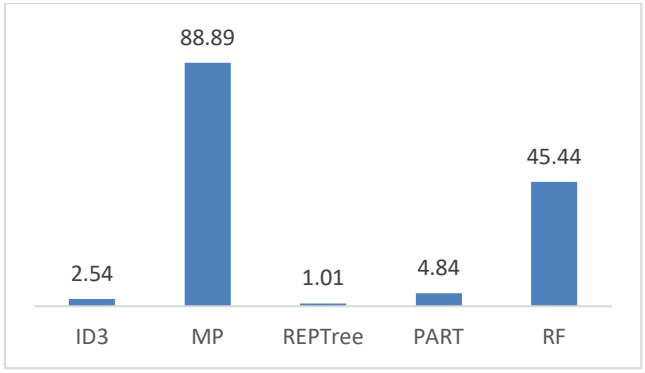


Fig. 6. Training Time Using Set 4

Because of the results in Table 4, the rest of the paper will focus on the results gained by using features in Set 2 and Set 3 only.

ii. Accuracy: The analysis of the data reveals that four classifiers, J48, RepTree, PART, and Random Forest have an overall accuracy greater than 99.9% when using features in set 2, and set 3. RF gets the highest accuracy with 99.99% in both feature sets as presented in Table 5.

Table 5
Overall accuracy of the classifiers after implementation using features in Set 1 and Set 2

Algorithm	Overall Accuracy	
	Set 2	Set 3
J48	99.9684	99.9647
MP	99.3122	98.1159
REPTree	99.9566	99.955
PART	99.9712	99.965
RF	99.9909	99.9903

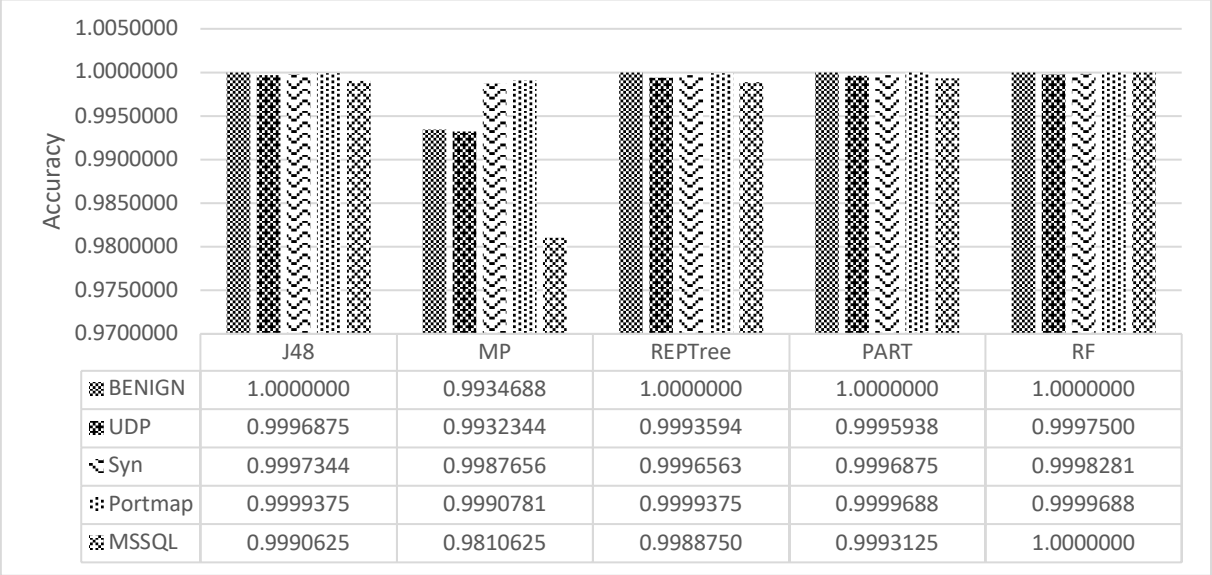
When diving deep through the data, the numbers show that all the classifiers successfully detected the BENIGN with 100% of accuracy except MP. Portmap got the second-best detection result with an accuracy average of 99.9% by all the classifiers in both sets, in contrast, the MSSQL attack had the highest error rate in all the classifiers' results except with the RF classifier which classified it with 100% accuracy. All the other classifiers have acceptable results in most of the cases as illustrated in Table 6.

Table 6
Accuracy of classifying different DDoS attacks when implementing classifiers using features in Set 2 and Set 3

Algorithm	Attack	Accuracy	
		Set 2	Set 3
J48	BENIGN	1	1
	UDP	0.9996875	0.999609
	Syn	0.99973438	0.999734
	Portmap	0.9999375	0.999891
	MSSQL	0.9990625	0.999
MP	BENIGN	0.99346875	0.989781
	UDP	0.99323438	0.98325
	Syn	0.99876563	0.993734
	Portmap	0.99907813	0.999156
	MSSQL	0.9810625	0.939875
REPTree	BENIGN	1	1
	UDP	0.99935938	0.999234
	Syn	0.99965625	0.99975
	Portmap	0.9999375	0.999875
	MSSQL	0.998875	0.998891
PART	BENIGN	1	1
	UDP	0.99959375	0.999516
	Syn	0.9996875	0.999734
	Portmap	0.99996875	0.999906
	MSSQL	0.9993125	0.999094
RF	BENIGN	1	1
	UDP	0.99975	0.999781

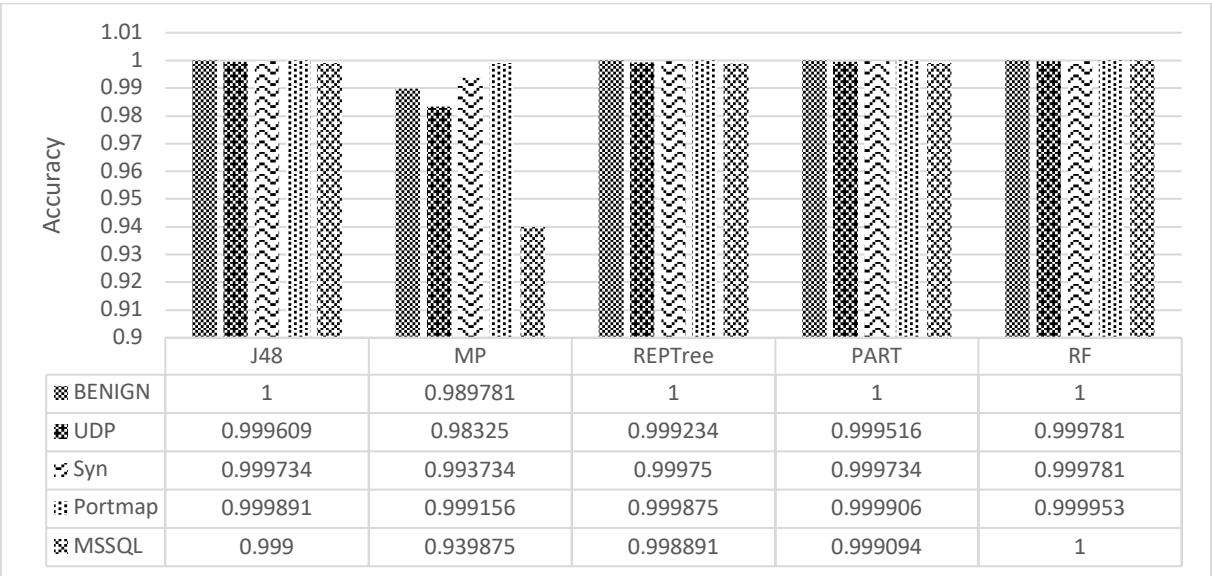
Syn	0.99982813	0.999781
Portmap	0.99996875	0.999953
MSSQL	1	1

RF, PART, and J48 respectively had the best accuracy among all the classifiers with minor differences not exceeding 0.01 to 0.02 when using features of set 1 and set 2. Figure 7 and Figure 8 illustrates the accuracy differences between the classifiers graphically for a better explanation.



	J48	MP	REPTree	PART	RF
BENIGN	1.0000000	0.9934688	1.0000000	1.0000000	1.0000000
UDP	0.9996875	0.9932344	0.9993594	0.9995938	0.9997500
Syn	0.9997344	0.9987656	0.9996563	0.9996875	0.9998281
Portmap	0.9999375	0.9990781	0.9999375	0.9999688	0.9999688
MSSQL	0.9990625	0.9810625	0.9988750	0.9993125	1.0000000

Fig. 7. Accuracy of classifiers based on attack type using features of Set 2

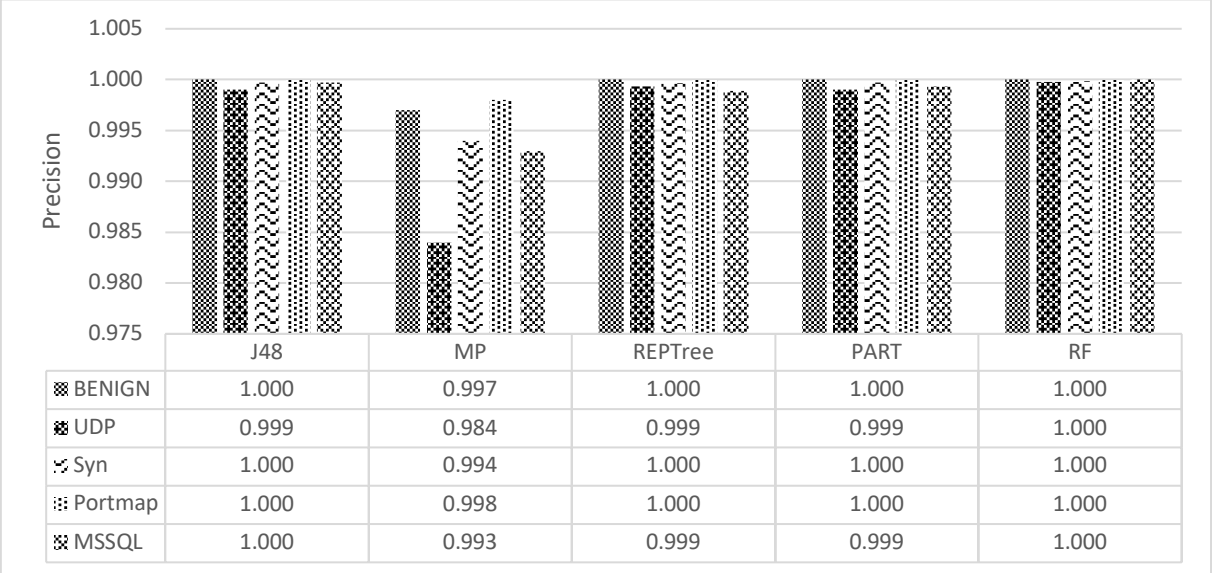


	J48	MP	REPTree	PART	RF
BENIGN	1	0.989781	1	1	1
UDP	0.999609	0.98325	0.999234	0.999516	0.999781
Syn	0.999734	0.993734	0.99975	0.999734	0.999781
Portmap	0.999891	0.999156	0.999875	0.999906	0.999953
MSSQL	0.999	0.939875	0.998891	0.999094	1

Fig. 8. Accuracy of classifiers based on attack type using features of Set 3

iii. Precision: effective classifier should ideally have a precision value of 1 Only when the numerator and denominator are equal, i.e. TP = TP + FP, does precision equal 1, and this also implies that FP is zero. As illustrated in Figure 9, when using Set 2, the majority of the precision values are (1) except in the MLP classifier. The Precision dropped to 0.9 in classifying UDP attacks by the classifiers (J48, MP, REPTree, and PART). Excluding J48 the 0.9 precision is still gained when classifying the MSSQL attack with the rest classifiers. RF

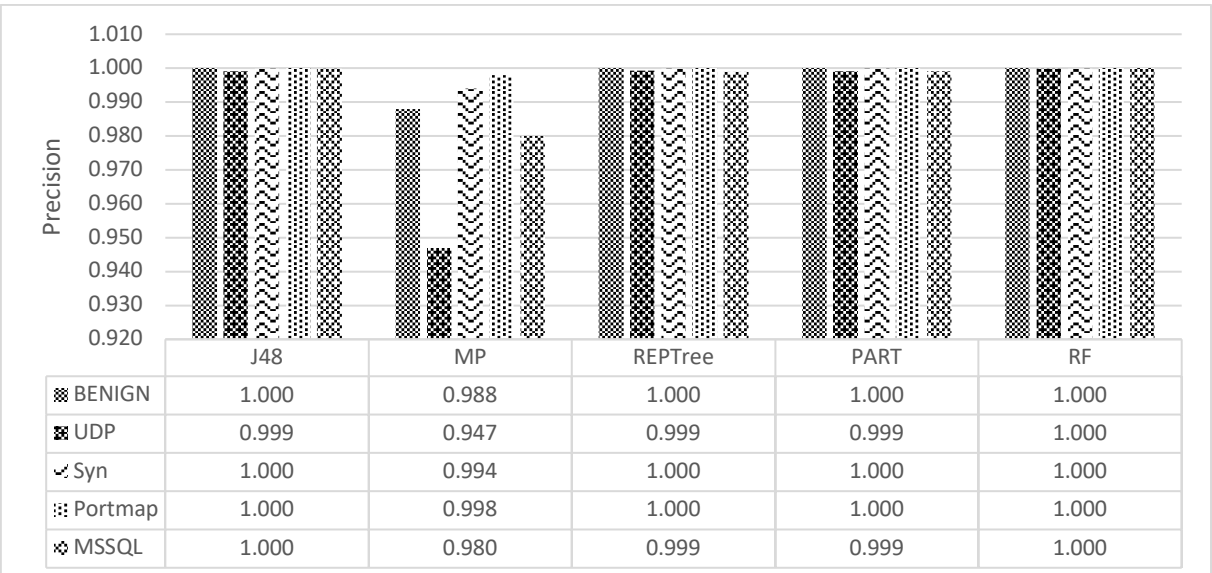
is the only classifier that had precision with value (1) when classifying all the attack cases. The results indicate that when using precision as a performance metric the better classifier is RF followed by J48. On the other hand, the values which have a precision of 0.9 means that FP is 0.1 or less which is acceptable, not bad.



	J48	MP	REPTree	PART	RF
BENIGN	1.000	0.997	1.000	1.000	1.000
UDP	0.999	0.984	0.999	0.999	1.000
Syn	1.000	0.994	1.000	1.000	1.000
Portmap	1.000	0.998	1.000	1.000	1.000
MSSQL	1.000	0.993	0.999	0.999	1.000

Fig. 9. Precision of classifiers based on attack type using features of Set 2

Figure 10, shows that when using Set 3, RF is still the only classifier that had (1) when classifying all the attack cases, also all the other cases are similar to the results that were gained when implementing the classifiers using Set 2 features.

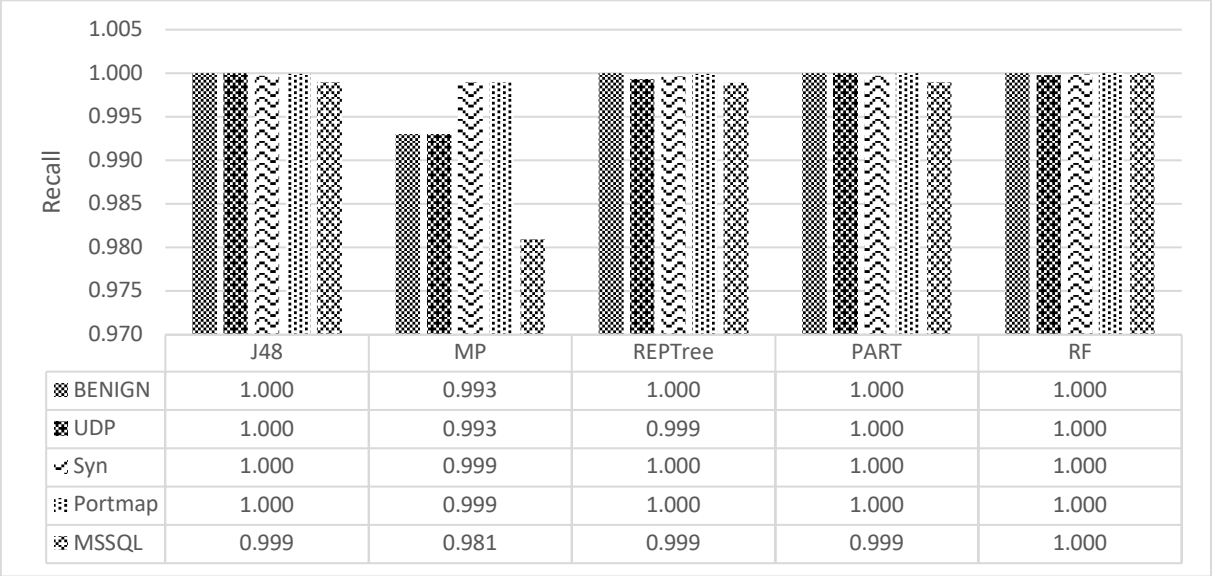


	J48	MP	REPTree	PART	RF
BENIGN	1.000	0.988	1.000	1.000	1.000
UDP	0.999	0.947	0.999	0.999	1.000
Syn	1.000	0.994	1.000	1.000	1.000
Portmap	1.000	0.998	1.000	1.000	1.000
MSSQL	1.000	0.980	0.999	0.999	1.000

Fig. 10. Precision of classifiers based on attack type using features of Set 3

iv. Recall: The recall is calculated by dividing the number of Positive samples correctly classified as Positive by the total number of Positive samples. The recall reflects the classifier's ability to recognize positive samples. More affirmative samples are detected as recall increases. In using Set 2, the classification results show that also in recall RF is still

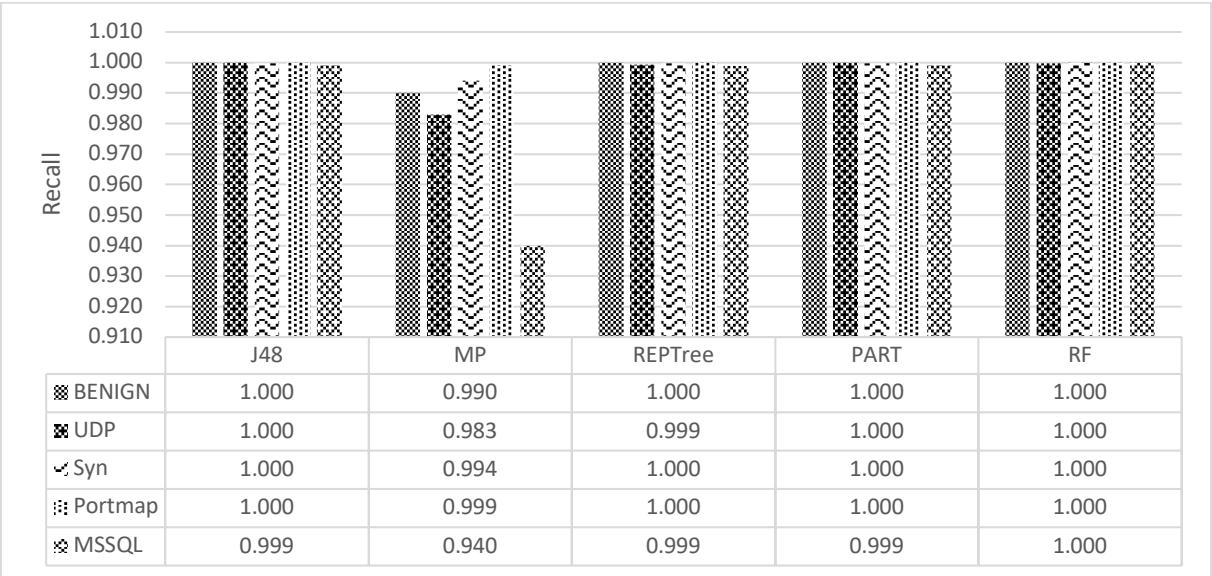
the best classifier followed by J48 and PART where REPTree got the fourth place and MLP the last. See Figure 11.



	J48	MP	REPTree	PART	RF
BENIGN	1.000	0.993	1.000	1.000	1.000
UDP	1.000	0.993	0.999	1.000	1.000
Syn	1.000	0.999	1.000	1.000	1.000
Portmap	1.000	0.999	1.000	1.000	1.000
MSSQL	0.999	0.981	0.999	0.999	1.000

Fig. 11. Recall of classifiers based on attack type using features of Set 2

The same results were gained when using Set 3 features and the classifiers still have the same performance sequence see Figure 12.



	J48	MP	REPTree	PART	RF
BENIGN	1.000	0.990	1.000	1.000	1.000
UDP	1.000	0.983	0.999	1.000	1.000
Syn	1.000	0.994	1.000	1.000	1.000
Portmap	1.000	0.999	1.000	1.000	1.000
MSSQL	0.999	0.940	0.999	0.999	1.000

Fig. 12. Recall of classifiers based on attack type using features of Set 3

By analysing the results of four performance measures obtained by the five classifiers for each of the two feature sets, it is determined that Set 2 obtains the highest overall performance of the classifiers. The utmost accuracy of the Random Forest classifier was 99.99% with both sets. PART and J48 classifiers detected attacks more accurately with set 2 which contains 6 features (Source Port, Destination Port, Flow Bytes/s, Flow Packets/s, Packet Length Mean, Packet Length Std), both obtained 99.9712 and 99.684 percent accuracy respectively.

The rest of the classifiers are not bad but the mentioned three had the highest accuracy score in identifying 4 types of DDoS Attacks (UDP, SYN, Portmap, and MSSQL). Because of the minor difference between the three classifiers in accuracy, precision, and recall, the training time was used to give a final decision. The fastest trained classifier in implementation was REPTree with 0.96 seconds when using set 3, followed by J48 with 2.23 seconds, and 4.28 seconds by PART with the same set of features. The same classifiers (REPTree, J48, and PART) with the same sequence, were trained using set 2 in 1.44, 2.8, and 5.74 seconds respectively. In contrast, the classifier with the highest time consumption was MLP and RF respectively in both sets. To determine the better classifier, the best feature set should be determined first, according to the accuracy obtained, feature set 2 is better. Now to determine the better classifier which can be used as a classifier in a real-time proposed model, the time factor is very important to take into consideration. Because of the high time required to train, the highest accuracy classifier RF is excluded. the second highest accuracy (PART) classifier required time more than J48 which has approximately the same accuracy score as PART see Table 7. For that, the decision was taken to use J48 as a classifier with acceptable accuracy, precision, recall, and training time. The highest accuracy score for each feature group is summarized in Table 7.

Table 7
The best accurate classifiers with feature sets and training time

Set No.	The classifiers	The Accuracy scores %	Train time In seconds
Set 2	Random Forest (RF)	99.990	56.14
	PART	99.971	5.74
	J48	99.964	2.8
Set 3	Random Forest (RF)	99.990	58.57
	PART	99.965	4.28
	J48	99.964	2.23

Table 8 shows the accuracy of several previous experiments in comparison with this study.

Table 8
A comparison between this study and existing systems

#	Authors	Year of the study	Detection Accuracy	Features
1	[25]	2020	99.79	22
2	[26]	2020	99.55%	10
3	[27]	2021	99.79%	15
			96.47%	4
4	[28]	2021	99.50%	unknown
5	[29]	2022	99.51	4
			99.96	8
6	This study	2023	99.97	6

The proposed model with six features (selected based on wrapper and information gain) produced the best results for identifying DDoS attacks compared to previous research.
Experiments indicate that the proposed J48 classifier requires less training time and performs better with an accuracy of approximately 99.97%, as shown in Table 8 for earlier works.

6. Conclusion

Using the CICDDoS2019 dataset, five supervised machine learning algorithms were implemented to determine its efficiency in detecting 4 types of DDoS attacks utilizing performance measures like training time, accuracy, precision, and recall. By wrapper and information gain, 4 sets of features are determined. The more efficient set was the one that contained six features. All the classifiers achieved good and acceptable accuracy, precision, and recall, but for fast detection, training time is used as an additional factor.

Experiments demonstrated that the RF algorithm was the most accurate with an accuracy rate of 99.99% but the training time was very high. In contrast, the J48 algorithm accuracy reached 99.97% in 2.8 seconds. Because of That, J48 is the best option for detecting 4 types of DDoS attacks when using the six selected features in Set 2.

Acknowledgement
This research was not funded by any grant.

References

[1] Arshi, M., M. D. Nasreen, and Karanam Madhavi. "A survey of DDoS attacks using machine learning techniques." In *E3S Web of Conferences*, vol. 184, p. 01052. EDP Sciences, 2020. https://doi.org/10.1051/e3sconf/202018401052

[2] Alsirhani, Amjad, Srinivas Sampalli, and Peter Bodorik. "Ddos detection system: utilizing gradient boosting algorithm and apache spark." In *2018 IEEE Canadian Conference on Electrical & Computer Engineering (CCECE)*, pp. 1-6. IEEE, 2018. https://doi.org/10.1109/CCECE.2018.8447671

[3] Ambusaidi, Mohammed A., Xiangjian He, Priyadarsi Nanda, and Zhiyuan Tan. "Building an intrusion detection system using a filter-based feature selection algorithm." *IEEE transactions on computers* 65, no. 10 (2016): 2986-2998. https://doi.org/10.1109/TC.2016.2519914

[4] Aljawarneh, Shadi, Monther Aldwairi, and Muneer Bani Yassein. "Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model." *Journal of Computational Science* 25 (2018): 152-160. https://doi.org/10.1016/j.jocs.2017.03.006

[5] Shameli-Sendi, Alireza, Makan Pourzandi, Mohamed Fekih-Ahmed, and Mohamed Cheriet. "Taxonomy of distributed denial of service mitigation approaches for cloud computing." *Journal of Network and Computer Applications* 58 (2015): 165-179. https://doi.org/10.1016/j.jnca.2015.09.005

[6] Canadian Institute for Cybersecurity. "DDoS 2019 Datasets Research UNB." (2019). https://www.unb.ca/cic/datasets/ddos-2019.html

[7] Song, Ruoning, and Fang Liu. "Real-time anomaly traffic monitoring based on dynamic k-NN cumulative-distance abnormal detection algorithm." In *2014 IEEE 3rd International Conference on Cloud Computing and Intelligence Systems*, pp. 187-192. IEEE, 2014. https://doi.org/10.1109/CCIS.2014.7175727

[8] Jia, Bin, Xiaohong Huang, Rujun Liu, and Yan Ma. "A DDoS attack detection method based on hybrid heterogeneous multiclassifier ensemble learning." *Journal of Electrical and Computer Engineering* 2017 (2017). https://doi.org/10.1155/2017/4975343

[9] Nezhad, Seyyed Meysam Tabatabaie, Mahboubeh Nazari, and Ebrahim A. Gharavol. "A novel DoS and DDoS attacks detection algorithm using ARIMA time series model and chaotic system in computer networks." *IEEE Communications Letters* 20, no. 4 (2016): 700-703. https://doi.org/10.1109/LCOMM.2016.2517622

[10] Prasad, K. Munivara, A. Rama Mohan Reddy, and K. Venugopal Rao. "Discriminating DDoS attack traffic from flash crowds on Internet Threat Monitors (ITM) using entropy variations." *African Journal of Computing & ICT* 6, no. 2 (2013): 53-62.

[11] Badis, Hammi, Guillaume Doyen, and Rida Khatoun. "Understanding botclouds from a system perspective: A principal component analysis." In *2014 IEEE Network Operations and Management Symposium (NOMS)*, pp. 1-9. IEEE, 2014. https://doi.org/10.1109/NOMS.2014.6838310

[12] Palagiri, Chandrika, Rasheda Smith, and Alan Bivens. "Network-based intrusion detection using neural networks." *Department of Computer Science Rensselaer Polytechnic Institute Troy, New York* (2002): 12180-3590.

[13] Fouladi, Ramin Fadaei, Cemil Eren Kayatas, and Emin Anarim. "Frequency based DDoS attack detection approach using naive Bayes classification." In *2016 39th International Conference on Telecommunications and Signal Processing (TSP)*, pp. 104-107. IEEE, 2016. https://doi.org/10.1109/TSP.2016.7760838

[14] Ebtihal, Sameer Alghoson, and Onytra Abbass. "Detecting Distributed Denial of Service Attacks using Machine Learning Models." *International Journal of Advanced Computer Science and Applications* 12, no. 12 (2021). https://doi.org/10.14569/IJACSA.2021.0121277

[15] Kushwah, Gopal Singh, and Virender Ranga. "Optimized extreme learning machine for detecting DDoS attacks in cloud computing." *Computers & Security* 105 (2021): 102260. https://doi.org/10.1016/j.cose.2021.102260

[16] Elsayed, Mahmoud Said, Nhien-An Le-Khac, Soumyabrata Dev, and Anca Delia Jurcut. "Ddosnet: A deep-learning model for detecting network attacks." In *2020 IEEE 21st International Symposium on" A World of Wireless, Mobile and Multimedia Networks"(WoWMoM)*, pp. 391-396. IEEE, 2020. https://doi.org/10.1109/WoWMoM49955.2020.00072

[17] Bolodurina, I., A. Shukhman, D. Parfenov, A. Zhigalov, and L. Zabrodina. "Investigation of the problem of classifying unbalanced datasets in identifying distributed denial of service attacks." In *Journal of Physics: Conference Series*, vol. 1679, no. 4, p. 042020. IOP Publishing, 2020. https://doi.org/10.1088/1742-6596/1679/4/042020

[18] Kousar, Heena, Mohammed Moin Mulla, Pooja Shettar, and D. G. Narayan. "Detection of DDoS attacks in software defined network using decision tree." In *2021 10th IEEE International Conference on Communication Systems and Network Technologies (CSNT)*, pp. 783-788. IEEE, 2021. https://doi.org/10.1109/CSNT51715.2021.9509634

[19] Aytaç, Tuğba, Muhammed Ali Aydın, and Abdül Halim Zaim. "Detection DDOS attacks using machine learning methods." (2020).

[20] Cil, Abdullah Emir, Kazim Yildiz, and Ali Buldu. "Detection of DDoS attacks with feed forward based deep neural network model." *Expert Systems with Applications* 169 (2021): 114520. https://doi.org/10.1016/j.eswa.2020.114520

[21] Goparaju, Bhargavi, and Bandla Sreenivasa Rao. "Distributed Denial-of-Service (DDoS) Attack Detection using 1D Convolution Neural Network (CNN) and Decision Tree Model." *Journal of Advanced Research in Applied Sciences and Engineering Technology* 32, no. 2 (2023): 30-41. https://doi.org/10.37934/araset.32.2.3041

[22] Hall, Mark, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. "The WEKA data mining software: an update." *ACM SIGKDD explorations newsletter* 11, no. 1 (2009): 10-18. https://doi.org/10.1145/1656274.1656278

[23] UCLA Academic Planning and Budget. "Common Data Set" https://apb.ucla.edu/campus-statistics/common-data-set

[24] Handelman, Guy S., Hong Kuan Kok, Ronil V. Chandra, Amir H. Razavi, Shiwei Huang, Mark Brooks, Michael J. Lee, and Hamed Asadi. "Peering into the black box of artificial intelligence: evaluation metrics of machine learning methods." *American Journal of Roentgenology* 212, no. 1 (2019): 38-43. https://doi.org/10.2214/AJR.18.20224

[25] Stiawan, Deris, Mohd Yazid Bin Idris, Alwi M. Bamhdi, and Rahmat Budiarto. "CICIDS-2017 dataset feature analysis with information gain for anomaly detection." *IEEE Access* 8 (2020): 132911-132921. https://doi.org/10.1109/ACCESS.2020.3009843

[26] Çakmakçı, Salva Daneshgadeh, Thomas Kemmerich, Tarem Ahmed, and Nazife Baykal. "Online DDoS attack detection using Mahalanobis distance and Kernel-based learning algorithm." *Journal of Network and Computer Applications* 168 (2020): 102756. https://doi.org/10.1016/j.jnca.2020.102756

[27] Kurniabudi, Kurniabudi, Deris Stiawan, Darmawijoyo Darmawijoyo, Mohd Yazid Bin Idris, Bedine Kerim, and Rahmat Budiarto. "Important features of CICIDS-2017 Dataset for anomaly detection in high dimension and imbalanced class dataset." *Indonesian Journal of Electrical Engineering and Informatics (IJEEI)* 9, no. 2 (2021): 498-511. https://doi.org/10.52549/ijeei.v9i2.3028

[28] Swe, Yin Mon, Pye Pye Aung, and Aye Su Hlaing. "A slow ddos attack detection mechanism using feature weighing and ranking." In *Proceedings of the International Conference on Industrial Engineering and Operations Management*, pp. 4500-4509. 2021. https://doi.org/10.46254/AN11.20210797

[29] Kareem, Mohammed Ibrahim, and Mahdi Nsaif Jasim. "Fast and accurate classifying model for denial-of-service attacks by using machine learning." *Bulletin of Electrical Engineering and Informatics* 11, no. 3 (2022): 1742-1751. https://doi.org/10.11591/eei.v11i3.3688