Applied Information Technology And Computer Science Vol. 4 No. 1 (2023) 109-123

AITCS

Homepage: http://publisher.uthm.edu.my/periodicals/index.php/aitcs
e-ISSN :2773-5141

A Comparison of Six Machine Learning Techniques for Cloud DDoS Attack Detection

Thanadoln Boonsiri[1], Cik Feresa Mohd Foozy[1]*

[1]Faculty of Computer Science and Information Technology,
Universiti Tun Hussein Onn Malaysia, Parit Raja, Batu Pahat, 86400, Johor, MALAYSIA

*Corresponding Author Designation

DOI: https://doi.org/10.30880/aitcs.2023.04.01.007
Received 27 September 2022; Accepted 27 May 2023; Available online 30 June 2023

Abstract: Cloud computing is a network access approach that allows for convenient, limitless, on-demand network access to a public computer resource pool. The DDoS assault is one of the most serious risks to cloud users since it jeopardizes cloud providers' services and renders them inaccessible to legitimate clients. In other related works, the comprehensive comparison of machine learning techniques is only limited to one or two algorithms. Machine learning approaches are capable of detecting DDoS assaults as well as preventing them. CICDDoS2019 dataset is used in this work. This research involves four phases which are pre-processing, feature selection, classification, and parameter evaluation. The six machine learning techniques implemented in this research are Logical Regression, Random Forest, Support Vector Machines, Decision Tree, Naive Bayes and K-Nearest Algorithms. To evaluate the effectiveness of each machine learning algorithm, accuracy, precision, and recall are the parameters used. It is found that Random Forest, Support Vector Machines and K-Nearest Algorithms produce the best results in terms of accuracy, precision, and recall.

Keywords: Cloud, DDoS, Machine Learning

1. Introduction

Cloud computing is a constantly evolving technology that faces various security problems. The DDoS attack is one of the most well-known and damaging cyber-attacks in recent memory [1]. The goal of launching the DDoS assault is to deplete the victim's resources. SYN flood attacks, Network Type Protocol (NTP) amplification, Ping-of-Death (PoD), User Datagram Protocol (UDP) flood, HyperText Transfer Protocol (HTTP) flood, Domain Name System (DNS) flood and Zero-day attacks are the most common types of DDoS attacks.

A Denial-of-Service attack, in which the effectiveness of a system, server, web-based application, or web-based service is purposefully and intentionally affected, or the system becomes completely inaccessible, is possibly the most well-known attack that influences availability. A Distributed Denial

*Corresponding author: feresa@uthm.edu.my

of Service (DDoS) assault is the most serious threat to the IT industry [2], and it is becoming more common every year. Machine learning-based approaches mentioned in the literature are categorized by picking the greatest number of characteristics contained in the dataset. Thus, this framework will examine the features of the selected dataset and classification techniques to enhance the accuracy. Existing related works of detecting DDoS attacks in cloud using machine learning techniques have not considered using a variety of machine learning techniques and testing their efficiency against one another. This research aims to test and validate the six machine learning algorithms by using Accuracy, Precision and Recall.

The objectives of this research are to propose a cloud DDoS attack detection framework using machine learning techniques, to identify suitable features of cloud DDoS attack detection framework using machine learning, to test and validate the framework by using Accuracy, Precision and Recall

The dataset used in this research project is the CICDDoS2019 dataset. CICDDoS2019 is a collection of harmless and up-to-date typical DDoS assaults that closely mimics real-world data (PCAPs). In this research, the NTP attack class and benign attacks will be used to test the efficiency of the machine learning algorithms.

The machine learning algorithms used in this research are Logical Regression, Random Forest, Support Vector Machines, Decision Tree, Naive Bayes and K-Nearest Algorithms. The software used to test the machine learning algorithms is Weka.

In this project, a cloud DDoS attack detection framework using machine learning techniques will be proposed. By accomplishing this objective, this research will redound to society's benefits, considering that cloud computing plays a crucial role in the world of information technology and DDoS attacks have the potential to render this technology useless and cause havoc. This research hopes to identify and analyze which framework or algorithm is best used to detect DDoS attacks in cloud computing.

2. Related Work

In this section, DDoS attacks, machine learning techniques and other related work will be analyzed and reviewed.

2.1 DDoS Attacks

A distributed denial-of-service (DDoS) attack is a malicious attempt to interrupt a specific server's, service's, or network's regular traffic by flooding the target or its surrounding infrastructure with Internet traffic. DDoS attacks are efficient because they use numerous hacked computer systems to attack traffic sources. DDoS assaults are continually changing as the aspects of technology utilized and the attackers' intentions change [3].

According to a survey done by Mahjabin et al [4], NTP (Network Time Protocol) amplification attack is a type of bandwidth depletion attack. This sort of attack is when the attacker's objective is to drain all the network bandwidths on the victim's machine using an attack army. This results to the victim denying access to real users for a certain amount of time until the attack is resolved.

2.2 DDoS Attacks in a cloud computing environment

The use of hardware and software to deliver services to end users across a network such as the internet is known as cloud computing. It consists of a collection of virtual machines that act as stand-ins for actual computers and deliver services like operating systems and apps. In a cloud computing environment, where resources are shared by multiple users, DDoS assaults are a huge security problem. Dealing with DDoS assaults at all tiers in cloud systems is tough since it's difficult to tell the difference

between the attacker's demands and genuine user requests, especially when the latter come from a huge number of distributed workstations [5].

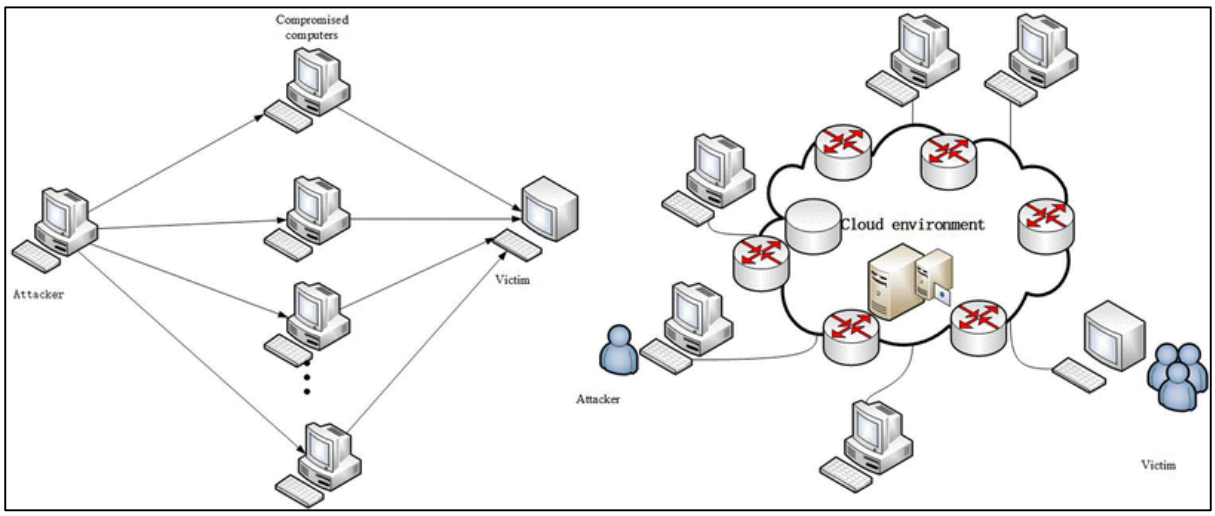

Figure 1: Traditional DDoS attack and the DDoS attack in cloud environment [6]

2.3 Cloud DDoS attack detection techniques

There are various techniques and methods in order to detect DDoS attacks in a cloud computing environment. Signature-based, anomaly-based, and hybrid approaches are the three types of techniques that can be used. Some of these methods include the use of artificial intelligence and machine learning.

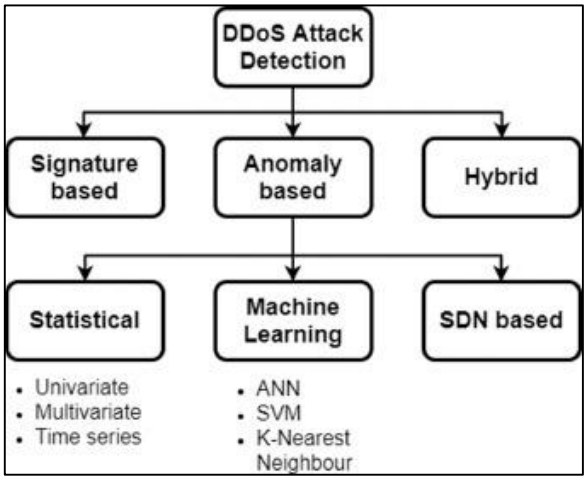

Figure 2: Taxonomy of various techniques towards DDoS attack detection [7]

2.3.1 Artificial intelligence approach in cloud DDoS detection

In a research proposed to detect recognized and unrecognized DDoS attacks, an anomaly-based detection system with a signature-based detection system was presented using an integrated artificial neural network [8]. The development of a distributed neural network was proposed to detect the unrecognized DDoS attack based-anomaly approach. The proposed method's efficiency and accuracy are demonstrated by the results. The proposed approach has the potential to improve current IDS against DDoS attacks on cloud computing.

In a review of artificial intelligence in DDoS attack and defense methods, the findings indicate the effectiveness and accuracy of the suggested approach. The suggested technique has the potential to enhance current IDS for cloud computing DDoS assaults. The review provides a comprehensive and

extensive examination of statistics and artificial intelligence technologies for identifying and countering DDoS assaults. It is found that defensive solutions that use statistical and artificial intelligence approaches perform better against DDoS attacks.

2.4 Machine learning approach in cloud DDoS detection

There are various machine learning algorithms. Some of the machine learning algorithms that will be reviewed in this section include Logical Regression, Random Forest, Support Vector Machines, Decision Tree, Naive Bayes and K-Nearest Algorithms.

2.4.1 Logical Regression

The supervised learning classification method logistic regression is used to predict the likelihood of a target variable. Because the nature of the goal or dependent variable is dichotomous, there are only two classifications. In basic terms, the dependent variable is binary in nature, with data represented as 1 (representing success/yes) or 0 (representing failure/no). A logistic regression model predicts P(Y=1) as a function of X mathematically. It is one of the most basic machine learning algorithms that may be used to a variety of categorization tasks [9] as in Equation 1

$$y = e^{(b0 + b1*x)} / (1 + e^{(b0 + b1*x)}) \qquad Eq.1$$

where y is the projected output, b0 is the bias or intercept term, and b1 is the single input value coefficient (x). Each column in the input data has a corresponding b coefficient (a constant real value) that must be determined using training data.

2.4.2 Random Forest

Random forest is a supervised learning technique that may be used to classify and predict data. However, it is mostly employed to solve categorization issues. A forest, is made up of trees, and more trees equals a more healthy forest. Similarly, the random forest method constructs decision trees from data samples, extracts predictions from each, and then votes on the best option. It's an ensemble approach that's superior than a single decision tree since it averages the results to reduce over-fitting [10]. Random forest is used in the studies of Abdul Moqeet [11], Makkawi et al [12], and Wani et al [13] . Below is an illustration of random forest :

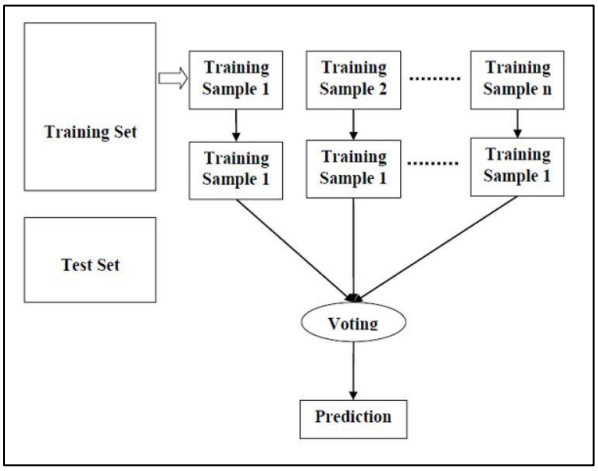

Figure 3: Random forest diagram [10]

2.4.3 Support Vector Machines (SVM)

SVMs are supervised machine learning techniques that may be used for both classification and regression. However, they are most commonly utilised in classification difficulties. In multidimensional space, an SVM model is essentially a representation of distinct classes in a hyperplane. SVM will

generate the hyperplane in an iterative way in order to reduce the error. SVM's purpose is to partition datasets into classes such that a maximum marginal hyperplane may be found [14]. Wani et al [13] implemented the technique of SVM in the study of DDoS detection.

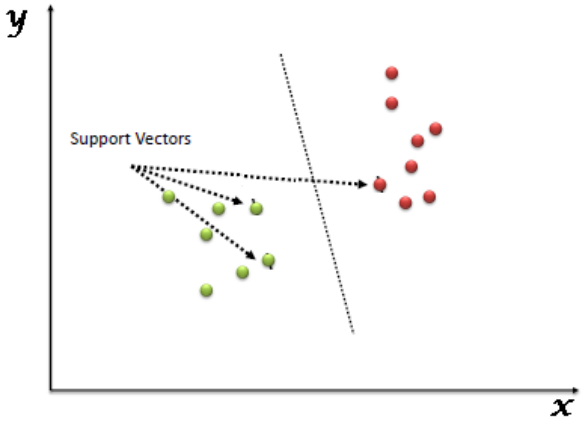

Figure 4: Classification by finding the hyper-plane that differentiates the two classes [20]

2.4.4 Decision Tree

Decision tree analysis is a prediction modelling technique that may be used in a variety of situations. An algorithmic strategy that can partition the information in numerous ways based on different circumstances can be used to create decision trees. The most powerful algorithms in the domain of supervised algorithms are decision trees [15]. Abdul Moqeet [11] implemented decision tree in order to classify TCP, UDP and ICMP protocols.

2.4.5 Naive Bayes

The Bayes theorem is used in Naive Bayes algorithms, which is a classification strategy based on the firm presumption that all predictors are independent of one another. To put it another way, the assumption is that the presence of a feature in a class is unrelated to the presence of other features in the same class. The basic goal of Bayesian classification is to determine the posterior probabilities, or the likelihood of a label given certain observed features, P(L | features) [16]. This can be expressed in the form in Equation 2. Wani et al [13] used Naive Bayes classifier in the study of DDoS attack detection.

$$P(L|features)=P(L)P(features|L)/P(features) \qquad Eq.2$$

2.4.6 K-Nearest Algorithms (KNN)

The KNN algorithm is a sort of supervised machine learning method that may be used to solve both classification and regression predicting problems. The KNN algorithm predicts the values of new data points based on 'feature similarity,' which implies that the new data point will be given a value depending on how close it resembles the points in the training set [17].

2.5 Cloud DDoS Attack Detection Techniques Using Machine Learning Algorithms

In this section, related works are reviewed in order to identify and compare datasets used, pre-processing phases, features selection process, classification phase as well as parameter evaluation process.

2.5.1 Dataset

In a related research done by Abdul Moqeet, the NSL-KDD dataset is used [11] . The NSL-KDD dataset has been used to evaluate the suggested machine learning technique's efficacy. The data set has 43

attributes per record, with 41 of them relating to the traffic input and the remaining two being labels which are whether it's a normal or attack and score being the severity of the traffic input itself.

Makkawi et al [12] used the UNSW-NB15 dataset. The dataset, which includes contemporary nine categories of assaults, was first released in 2015. Furthermore, the UNSW-NB15 dataset has 49 characteristics that cover the whole class label, totaling 2540044 entries. Content features, flow features, temporal features, basic features, extra produced features, and labelled features are the different types of features. Wani et al [13] utilized Tor Hammer as an attacking tool on the owncloud environment, and a fresh dataset was constructed using Intrusion Detection System.

2.5.2 Pre-processing

Abdul Moqeet [11] conduct the pre-processing phase of the NSL-KDD dataset by extracting necessary attributes from the input cloud network [17]. The attributes are derived from the traffic that is coming in. Packets include a variety of information regarding the qualities found in log files. Such characteristics aid in distinguishing between genuine and malicious traffic. To bring all characteristics to a regular scale [0 - 1], the preprocessing module use the minmax normalisation algorithm. The data is separated into training and testing datasets after normalisation and delivered to the next subsystem in order to continue the procedure.

Wani et al [19], take files from the Owncloud server were sent into the Intrusion Detection System SNORT. This open source rule-based programme is used to detect all of these assaults, although the default rules for detecting DDoS attacks have been altered. The SNORT output was controlled by specifying the needed tuples. The alert generated from SNORT consists of 9 features.

2.5.3 Features Selection

To obtain optimal system performance, it is critical to employ the smallest number of features possible [24]. This reduces the complexity of time and space. The attribute selection module is used to decrease the number of parameters. Abdul Moqeet compared two types of algorithms used for collection of attributes which are Filter method and Wrapper approach. Ultimately, the Filter approach was chosen for the project since it is quick, straightforward, and produces quick results[17]. The Correlation Feature Selection (CFS) Technique is being used for traffic filtering. Selection of features is an approach for removing irrelevant and unneeded characteristics from a dataset in order to improve learning accuracy and predictability of classifiers [25].

For the feature selection process of Makkawi et al [18], Flow Features, Basic Features, Content Features, Time Features, Additional Generated Features, and Labelled Features are the six groups that the features are categorised into. The General Purpose Features and Connection Features subgroups of Additional Generated Features are further divided. Wani et al [13] generated alerts from SNORT consisting of 9 features as listed in Table 1.

Table 1 : Dataset Features

Feature	Description
Duration	Duration of the flow
Proto	Type od protocol
Source Ip	Internet Protocol Address (source)
Dest IP	Internet Protocol Address (destination)
Src IP	Port (source)
DST Ip	Port (destination)
Packets	Transmitted Packets
Class	Attack classification labels
Bytes	Number of transmitted bytes

2.5.4 Classification

Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Internet Control Message Protocol (ICMP) flooding assaults may be detected using a machine learning-based defensive mechanism [11]. To recognize the different types of attacks, the dataset's protocol type characteristics (attributes) are used to evaluate the system. It's crucial to know the difference between TCP, UDP, and ICMP flooding assaults in a cloud computing context. Incoming traffic is classified based on its class characteristic (attribute). The protocol kinds are TCP, UDP, and ICMP. J48 and RF, the two best-chosen classifiers in a variety of circumstances, are employed.

Makkawi et al [12], investigated the recommended features and a few machine learning methods in Weka, including the Random Forest algorithm, were used to the UNSW-NB15 dataset, with the following subset of features outperforming the others. For data classification, Wani et al [13] looked into and evaluated three machine learning algorithms: Random, Forest, Naive Bayes, and Support Vector Machine. These algorithms were chosen due to their high efficiency and usefulness in the field of network security.

2.5.5 Parameter Evaluation

In order to assess the research's effectiveness, several parameters are used such as accuracy, true positive rate, true negative rate, precision, recall, F-score and RootMeanSquareError. Accuracy is the proportion of normal and anomaly classes properly detected in the provided dataset [11]. True positive rate is the proportion of normal class accurately detected in the provided dataset. True negative rate is the percentage of anomaly classes accurately detected in the provided dataset. Precision is used to detect relevant types of attacks rather than non specific types. Recall is an approach that detects specific types of DDoS attacks than actual ones. F-score is the harmonic mean of the model's precision and recall, and it is specified as the sum of the model's precision and recall. RootMeanSquareError is used to detect errors between actual and predicted classes. Makkawi et al [12], used accuracy in order to assess the research's effectiveness. Wani et al [13] gathered and evaluated using recall, precision, accuracy, specificity and F measure. Specificity is the effectiveness of a classifier to recognize negative labels. Table 2 shows the comparison of cloud DDOS detection using Machine Learning approach.

Table 2: Comparison of Cloud DDoS Detection Using Machine Learning Approach

	Abdul Moqeet [11]	Makkawi et al [12]	Wani et al [13]	Proposed research
Dataset	NSL-KDD	UNSW-NB15	Generated Dataset	CICDDoS2019
Pre-processing	Normalisation	Not mentioned	Alert processing and Mapping	Normalisation
Features Selection	src bytes, dst_bytes, logged_in, serror_rate, srv_rerror_rate, diff srv rate, srv diff host rate, dst host srv diff host rate, dst host srv error rate using Correlation Feature Selection (CFS) Technique	Content features, flow features, time features, basic features, additional generated features and labelled features	Duration, Proto, source IP, Dest IP, Src Pt, Dst Pt, Packets, class, Bytes	Content features, flow features, time features, basic features using Correlation Feature Selection (CFS) Technique
Classification	Decision tree and Random Forest	Random Forest	Random Forest, Naive Bayes and Support Vector Machine	Logical Regression, Random Forest, Support Vector Machines,

				Decision Tree, Naive Bayes, K-Nearest.

Table 2: (cont)

	Abdul Moqeet [11]	Makkawi et al [12]	Wani et al [13]	Proposed research
Classification	1. Decision tree 2. Random Forest	Random Forest	1. Random Forest 2. Naive Bayes 3. Support Vector Machine	1. Logical Regression 2. Random Forest 3. Support Vector Machines 4. Decision Tree 5. Naive Bayes 6. K-Nearest Algorithms
Parameter Evaluation	1. Accuracy, 2. True positive rate 3. True negative rate 4. Precision 5. Recall 6. F-score 7. RootMeanSquareError.	Accuracy only	1. Recall, 2. Precision 3. Accuracy 4. Specificity 5. F measure	1. Accuracy 2. Precision 3. Recall

3. Framework

There are five phases that needs to be implemented in order to achieve the objective of this research. Firstly, the CICDDoS2019 dataset will be downloaded from a trusted source that is the official website of The University of New Brunswick to whom the dataset belongs. Then, pre-processing of the dataset will be done to allow for the best features that are suitable for the detection of cloud DDoS attacks to be selected. After the feature selection process, the classification process will be conducted using six machine learning techniques by using the data with the selected features. Lastly, parameter evaluation will be done to determine the success of the machine learning techniques in detecting cloud DDoS attacks.

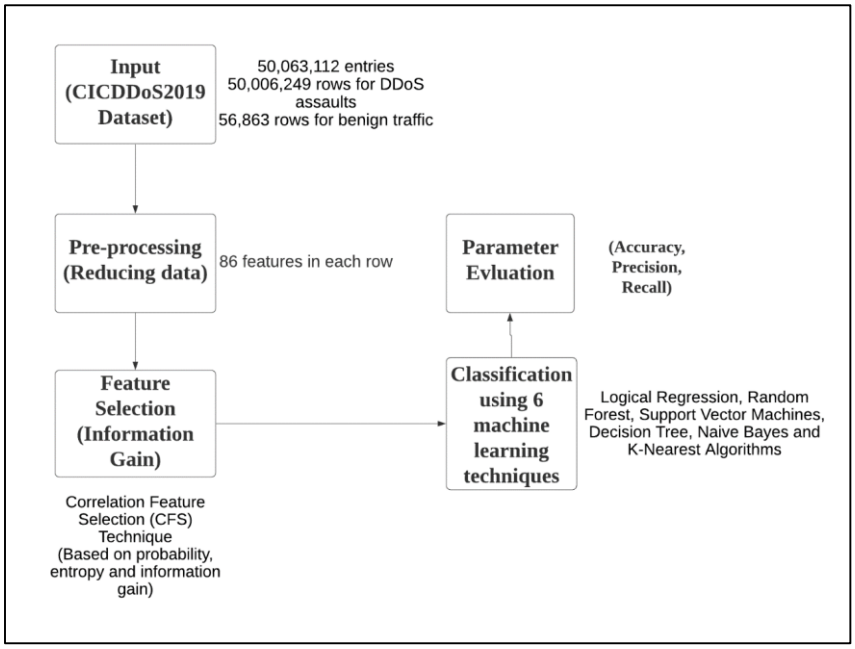

Figure 5: Research Framework

3.1 Dataset

The CICDDoS2019 dataset will be used taken from the official University of New Brunswick website. By utilizing this dataset, a clear picture of normal network traffic and DDoS attacks can be provided. There are 50,063,112 entries in the CIC-DDoS2019 dataset, including 50,006,249 rows for DDoS assaults and 56,863 rows for benign traffic. There are 86 features in each row [18].

3.2 Preprocessing

In the cloud system, log files store information about arriving and exiting packets. Arrived packets include both legitimate and malicious requests. These entries contain information such as sourceIPaddress, destinationIPaddress, label and many more. This information is important in the drawing of attributes that assist in the identification of an assault. Therefore, the relevant properties are collected from the incoming cloud network in the preprocessing subsystem. The characteristics are generated from this dataset and the values are standardized. Normalization is a scaling approach used in the pre-processing phase. Normalization is defined as the process of modifying original data without changing its behaviour or existence [11]. The features in the dataset are of various data formats and have varying values. As a result, in order to use machine learning and classifiers, all variables must be converted to a standard scale. The goal of normalization is to modify the values of the dataset's columns on a standard scale. To bring all features to a regular scale [0 - 1], the preprocessing module employs the minmax normalization algorithm. Following normalization, the data is divided into training and testing datasets.

3.3 Feature selection

Correlation Feature Selection (CFS) Technique is used for traffic filtration. CFS technique is based on probability, entropy, and information gain. A feature's probability is determined by whether a favorable situation exists. This probability is utilized to determine the relationship between each class attribute and the other attribute one by one. The entropy of an attribute(x) is computed in this fashion, indicating how closely these attributes are connected to the class attribute (x). A confidence matrix is created dynamically based on their relationship. In this data (subset of specified characteristics), an Information Gain formula is used, and if these values are equal to or greater than the confidence matrix value, it is chosen as the best subset among all other subsets.

3.4 Classification Techniques

In this research, 10-fold validation is used. Cross validation is a technique for calculating prediction error that is applied to a model and a data collection. It is perhaps the simplest and most extensively used approach for estimating prediction error [19]. 10-fold validation indicates that the entire data set is randomly partitioned into ten parts, nine of which are used to train the model and one of which is utilised for testing. This method is repeated ten times, with the error determined each time. The mean of the errors created in each iteration will be the model's overall error as illustrated in Table 3.

Table 3: Steps of 10-fold validation [20]

Iteration 1	Data randomly partitioned in 10 parts, 9 parts for training and 1 part for testing	Calculate error = E1
Iteration 2	Data randomly partitioned in 10 parts, 9 parts for training and 1 part for testing	Calculate error = E2
Iteration 3	Data randomly partitioned in 10 parts, 9 parts for training and 1 part for testing	Calculate error = E3
Iteration 4	Data randomly partitioned in 10 parts, 9 parts for training and 1 part for testing	Calculate error = E4

Table 3: (cont)

Iteration 5	Data randomly partitioned in 10 parts, 9 parts for training and 1 part for testing	Calculate error = E5
Iteration 6	Data randomly partitioned in 10 parts, 9 parts for training and 1 part for testing	Calculate error = E6
Iteration 7	Data randomly partitioned in 10 parts, 9 parts for training and 1 part for testing	Calculate error = E7
Iteration 8	Data randomly partitioned in 10 parts, 9 parts for training and 1 part for testing	Calculate error = E8
Iteration 9	Data randomly partitioned in 10 parts, 9 parts for training and 1 part for testing	Calculate error = E9
Iteration 10	Data randomly partitioned in 10 parts, 9 parts for training and 1 part for testing	Calculate error = E10
	Total Error = mean (E1,E2,E3,E4,E5,E6,E7,E8,E,9,E10)	

In Weka, the process of 10-fold validation could be done under the ‘classify’ tab. The results of the 10-fold validation is illustrated in the figure below.

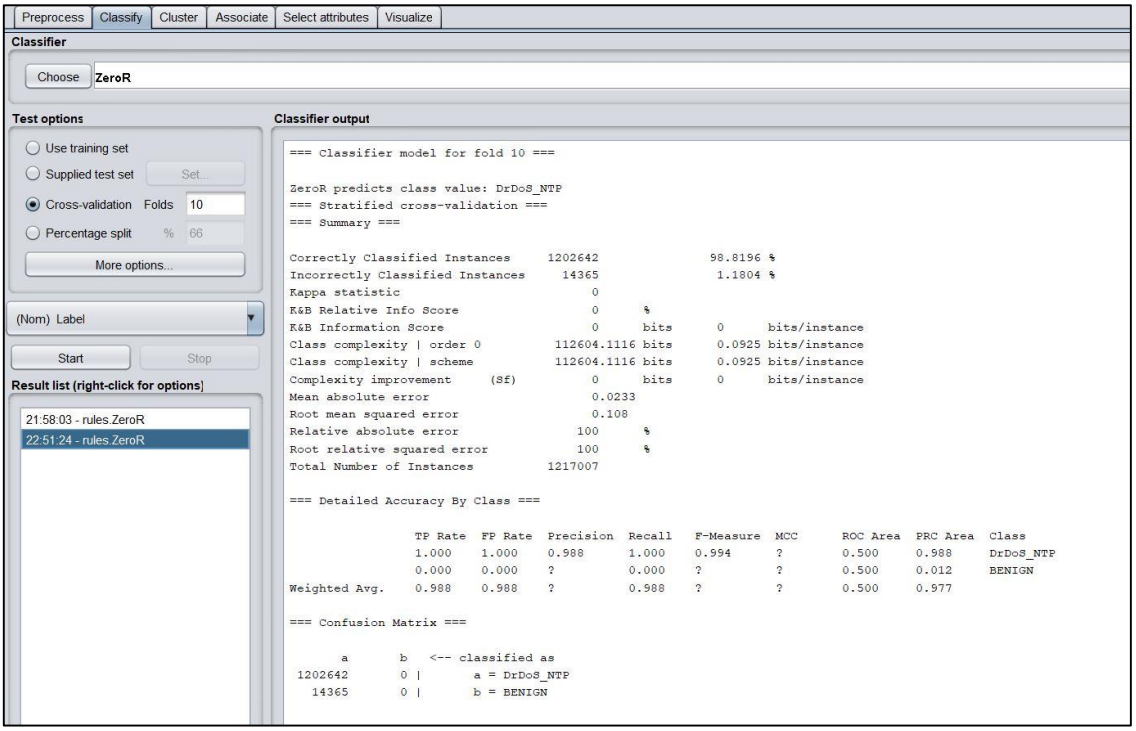

Figure 6: 10-fold validation using Weka

3.5 Hardware Requirement

The hardware used for this research is stated in Table 3.4. The specified hardware will be able to conduct analysis on cloud DDoS detection framework using six machine learning techniques.

Table 4: Hardware used for this research

Hardware	Specification
Acer Aspire 5 A515-51G-59ZO Laptop	Intel(R) Core(TM) i5-8250U CPU @ 1.60GHz 1.80 GHz 12GB DDR4 Memory 1000GB HDD Windows 10 Home Single Language 64-bit operating system, x64-based processor

4. Results and Discussion

This section describes the findings of this research. Experiments are done using the CICDDoS2019 dataset and Weka Explorer software tool.

4.1 Pre-processing

In the preprocessing phase, normalization is a scaling approach used and is defined as the process of modifying original data without changing its behaviour or existence. Since the features of the CICDDoS2019 dataset used in this research have different and varying ranges, to modify the values of the dataset's columns on a standard scale [0-1], the preprocessing module employs the minmax normalization algorithm.

4.2 Feature Selection

In order to know which features are relevant or otherwise in this research, correlation-based feature selection (CFS) will be used. The correlation between each attribute and the output variable could be calculated and attributes that have a moderate-to-high positive or negative correlation will be selected.

In Weka, the correlation-based feature selection with the CorrelationAttributeEval technique is used whereby the use of a Ranker search method is required. Using above discussed Correlation Feature Selection, the attributes that are most correlated to class attribute (benign/attack) are listed in Table 5.

Table 5: Attributes that are most correlated to class attribute

Rank	Feature
1	Source IP
2	Inbound
3	Bwd Packets
4	CWE Flag Count
5	Protocol
6	URG Flag Count
7	Min Packet Length
8	Fwd Packet Length Min
9	Source Port

4.3 Classification Result

To conduct the experiment for this research project the dataset will be partitioned into several smaller datasets. For instance, the experiment will be first conducted with all 86 features first followed by 56 features and 26 features using cross validation and percentage split. The experiment will also be conducted on the 9 selected features using both cross validation and percentage split.

Table 6: Classification using percentage split

No of features		Logical Regression	Random Forest	Support Vector Machines	Decision Tree	Naïve Bayes	K-Nearest Algorithms
86	Accuracy	99.75%	100%	100%	98.75%	65%	100%
	Precision	0.998	1	1	0.988	0.868	1
	Recall	0.998	1	1	0.988	0.65	1
56	Accuracy	90.75%	100%	85.25%	98.25%	35%	99.25%
	Precision	0.917	1	0.875	0.984	0.825	0.993
	Recall	0.908	1	0.853	0.983	0.35	0.993

Table 6: (cont)

No of features		Logical Regression	Random Forest	Support Vector Machines	Decision Tree	Naïve Bayes	K-Nearest Algorithms
26	Accuracy	83.75%	99.75%	85%	98%	28.25%	99.50%
	Precision	0.814	0.988	0.873	0.982	0.803	0.995
	Recall	0.838	0.988	0.85	0.98	0.283	0.995

Table 7: Classification using cross validation

No of features		Logical Regression	Random Forest	Support Vector Machines	Decision Tree	Naïve Bayes	K-Nearest Algorithms
86	Accuracy	100%	98.90%	100%	99.15%	65.68%	100%
	Precision	1	0.999	1	0.992	0.877	1

	Recall	1	0.999	1	0.991	0.657	1
	Accuracy	90.00%	99.70%	86.49%	98.90%	35.72%	98.65%
56	Precision	0.904	0.997	0.879	0.989	0.832	0.987
	Recall	0.9	0.997	0.865	0.989	0.357	0.986
	Accuracy	85.04%	99.60%	85.84%	98.60%	29.01%	99.40%
26	Precision	0.832	0.996	0.879	0.986	0.835	0.994
	Recall	0.85	0.996	0.858	0.986	0.29	0.994

Table 8: Classification on 10 selected features using cross validation

Classifier	Accuracy	Precision	Recall
Logical Regression	99.849%	0.999	0.998
Random Forest	100%	1.000	1.000
Support Vector Machines	100%	1.000	1.000
Decision Tree	99.699%	0.997	0.997
Naïve Bayes	91.395%	0.940	0.914
K-Nearest Algorithms	99.95%	1.000	0.999

Table 9: Classification on 10 selected features using 80% percentage split

Classifier	Accuracy	Precision	Recall
Logical Regression	99.5%	0.999	0.995
Random Forest	100%	1.000	1.000
Support Vector Machines	100%	1.000	1.000
Decision Tree	99.25%	0.993	0.993
Naïve Bayes	92.75%	0.948	0.928
K-Nearest Algorithms	100%	1.000	1.000

Table 6 demonstrates the results of the classification experiment conducted using percentage split. In the dataset with 86 features, the Random Forest, Support Vector Machines and K-Nearest Algorithms classifiers recorded perfect results in terms of accuracy, precision, and recall. In the dataset with 56 and 26 features, the Random Forest, Decision Tree, and K-Nearest Algorithms classifiers recorded the highest percentage of accuracy, precision and recall. Naïve Bayes classification recorded the worst percentage of accuracy, precision, and recall across all number of features.

Table 7 demonstrates the results of the classification experiment conducted using cross validation. In the dataset with 86 features, the Logical Regression, Support Vector Machines and K-Nearest Algorithm classifiers performed the best in terms of accuracy, precision and recall. In the dataset with 56 and 26 features, the Random Forest, Decision Tree and K-Nearest Algorithms classifiers recorded the highest percentage of accuracy, precision and recall. Naïve Bayes classification recorded the worst percentage of accuracy, precision and recall across all number of features.

Table 8 demonstrates the results of the classification experiment on the 9 selected features using cross validation. Random Forest and Support Vector Machines recorded the best performance in terms of accuracy, precision and recall. Naïve Bayes recorded the lowest accuracy with 91.395%, lowest precision with 0.940 and lowest recall with 0.914.

Table 9 demonstrates the results of the classification experiment on the 9 selected features using percentage split. Random Forest, K-Nearest Algorithms and Support Vector Machines classifiers

produced the best results with 100% accuracy, 1.000 precision and 1.000 recall. Naïve Bayes classifier produced the worst result with 92.75% accuracy, 0.948 precision and 0.928 recall values.

As compared to the related work of Abdul Moqeet [11], similar results in terms of accuracy, precision and recall of the Random Forest and Decision Tree classifiers across all experiments are found to be high. It is also found that the results of the Naïve Bayes algorithm in terms of accuracy, precision and recall are comparatively low compared to other machine learning algorithms.

5. Conclusion

Based on the results of classification using percentage split as well as cross validation on the dataset of varying number of features, Random Forest, Decision Tree and K-Nearest Algorithms classifiers produced the most consistent results with a high value of accuracy, precision and recall compared to Logical Regression, Support Vector Machines and Naïve Bayes classifiers.

Tables 8 and 9 show the results of the classification experiment on the 10 selected features vary from the results of the experiment done on the non-selected features. Using cross validation as well as percentage split, the Random Forest, Decision Tree and K-Nearest Algorithms classifiers produced the most consistent results with a high value of accuracy, precision and recall compared to Logical Regression, Support Vector Machines and Naïve Bayes classifiers.

The objective of this research is to design a cloud DDoS attack detection framework using machine learning techniques. This objective was achieved, and the detection framework consists of five phases. Firstly, the dataset was imported and the NTP attack class of the dataset was identified as the principal class. Next, the pre-processing phase of the dataset was conducted using normalization. The third phase, which was feature selection was orchestrated using the Correlation Feature Selection (CFS) Technique. Classification using 6 machine learning techniques was then conducted on the dataset and lastly the parameter evaluation phase was conducted based on accuracy, precision and recall.

The second objective, which was to identify suitable features of cloud DDoS attack detection framework and compare using machine learning was also successfully carried out. Out of the 86 features of the NTP class of the CICDDoS2019 dataset, the top 10 most correlated features were selected. The last objective of the research was to test and validate the framework by using Accuracy, Precision and Recall. This objective was completed as demonstrated by the results of the classification phase in Chapter 4.

This research's framework has several drawbacks and limitations. The experiments were only conducted on a single dataset which is the CICDDoS2019 dataset. As time goes by, more well known and latest datasets may be used to evaluate the efficiency of machine learning algorithms. Moreover, only the NTP attack class of the CICDDoS2019 dataset was used in the experiments. Therefore, there are a plethora of other types of DDoS attacks that might be investigated and analyzed in the future.

Acknowledgment

The authors would like to thank the Faculty of Computer Science and Information Technology, Universiti Tun Hussein Onn Malaysia for its support and encouragement throughout the process of conducting this research.

References

[1] O. Osanaiye, H. Cai, K.-K. R. Choo, A. Dehghantanha, Z. Xu, and M. E. Dlodlo, "Ensemble-based Multi-Filter Feature Selection Method for Distributed Denial-of-Service (DDoS) Attack Detection in Cloud Computing," EURASIP J. Wirel. Commun. Netw., vol. 2016, 2016, doi: 10.1186/s13638-016-0623-3.

[2] R. Maciel, J. Araujo, J. Dantas, C. Melo, E. Guedes, and P. Maciel, "Impact of a Distributed

Denial-of-Service (DDoS) Attack on computer systems: An approach based on an attack tree model," in 2018 Annual IEEE International Systems Conference (SysCon), Canada, 2018, pp. 1–8, doi: 10.1109/SYSCON.2018.8369611.

[3] J. Pei, Y. Chen, and W. Ji, "A Distributed Denial-of-Service (DDoS) Attack Detection Method Based on Machine Learning," vol. 1237, p. 32040, Jun. 2019, doi: 10.1088/1742-6596/1237/3/032040.

[4] T. Mahjabin, Y. Xiao, G. Sun, and W. Jiang, "A survey of distributed denial-of-service attack, prevention, and mitigation techniques," Int. J. Distrib. Sens. Networks, vol. 13, p. 155014771774146, 2017, doi: 10.1177/1550147717741463.

[5] M. Darwish, A. Ouda, and L. F. Capretz, "Cloud-based Distributed Denial-of-Service (DDoS) Attacks and defenses," in International Conference on Information Society (i-Society 2013), Canada, 2013, pp. 67–71.

[6] C. Wang, H. Yao, and Z. Liu, "An efficient Distributed Denial-of-Service (DDoS) Attack detection based on SU-Genetic feature selection," Cluster Comput., vol. 22, 2019, doi: 10.1007/s10586-018-2275-z.

[7] G. S. Kushwah and V. Ranga, "Voting extreme learning machine based distributed denial of service attack detection in cloud computing," J. Inf. Secur. Appl., vol. 53, p. 102532, 2020, doi: https://doi.org/10.1016/j.jisa.2020.102532.

[8] S. Alzahrani and L. Hong, "Detection of Distributed Denial of Service (DDoS) Attacks Using Artificial Intelligence on Cloud," in 2018 IEEE World Congress on Services (SERVICES), The United States of America, 2018, pp. 35–36, doi: 10.1109/SERVICES.2018.00031.

[9] X. Zou, Y. Hu, Z. Tian, and K. Shen, "Logistic Regression Model Optimization and Case Analysis," in 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT), China, 2019, pp. 135–139, doi: 10.1109/ICCSNT47585.2019.8962457.

[10] S. Sindhu, S. P. Patil, A. Sreevalsan, F. Rahman, and M. S. A. N., "Phishing Detection using Random Forest, Support Vector Machines and Neural Network with Backpropagation," in 2020 International Conference on Smart Technologies in Computing, Electrical and Electronics (ICSTCEE), India, 2020, pp. 391–394, doi: 10.1109/ICSTCEE49637.2020.9277256.

[11] M. Abdul, "A Machine Learning Based Classification Technique to Detect Distributed Denial-of-Service (DDoS) Attack in Cloud Computing Environment," Capital University of Science and Technology, 2021.

[12] A. M. Makkawi and A. Yousif, "Machine Learning for Cloud Distributed Denial-of-Service (DDoS) Attack Detection: A Systematic Review," in 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE), Sudan, 2021, pp. 1–9, doi: 10.1109/ICCCEEE49695.2021.9429678.

[13] A. R. Wani, Q. P. Rana, U. Saxena, and N. Pandey, "Analysis and Detection of Distributed Denial-of-Service (DDoS) Attack on Cloud Computing Environment using Machine Learning Techniques," in 2019 Amity International Conference on Artificial Intelligence (AICAI), United Arab Emirates, 2019, pp. 870–875, doi: 10.1109/AICAI.2019.8701238.

[14] L. Mohan, J. Pant, P. Suyal, and A. Kumar, "Support Vector Machine Accuracy Improvement with Classification," in 2020 12th International Conference on Computational Intelligence and Communication Networks (CICN), India, 2020, pp. 477–481, doi: 10.1109/CICN49253.2020.9242572.

[15] F.-J. Yang, “An Extended Idea about Decision Trees,” in 2019 International Conference on Computational Science and Computational Intelligence (CSCI), The United States of America, 2019, pp. 349–354, doi: 10.1109/CSCI49370.2019.00068.

[16] A. Kelly and M. A. Johnson, “Investigating the Statistical Assumptions of Naïve Bayes Classifiers,” in 2021 55th Annual Conference on Information Sciences and Systems (CISS), Online, 2021, pp. 1–6, doi: 10.1109/CISS50987.2021.9400215.

[17] A. Kumar, A. Verma, G. Shinde, Y. Sukhdeve, and N. Lal, “Crime Prediction Using K-Nearest Neighboring Algorithm,” in 2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE), India, 2020, pp. 1–4, doi: 10.1109/ic-ETITE47903.2020.155.

[18] I. Sharafaldin, A. H. Lashkari, S. Hakak, and A. A. Ghorbani, “Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy,” in 2019 International Carnahan Conference on Security Technology (ICCST), India, 2019, pp. 1–8, doi: 10.1109/CCST.2019.8888419.

[19] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, “The Elements of Statistical Learning: Data Mining, Inference, and Prediction,” Math. Intell., vol. 27, pp. 83–85, 2004, doi: 10.1007/BF02985802.