

Evaluation of DoS/DDoS Attack Detection with ML Techniques on CIC-IDS2017 Dataset

Saida Farhat¹^a, Manel Abdelkader², Amel Meddeb-Makhlouf¹^b and Faouzi Zarai¹^c

¹ENET'COM, NTS'COM Research Unit, University of Sfax, Tunisia

²Tunis Business School, University of Tunis, Tunisia

Keywords: Cloud Environment, Denial-of-Service (DoS/DDoS), Intrusion Detection, Machine Learning (ML), eXtreme Gradient Boosting (XGBoost).

Abstract: Cloud computing is one of today's most promising technologies. It provides its users with simplified IT infrastructure and management, remote access from effectively anywhere in the world with a stable internet connection, and cost efficiencies. Despite all these benefits, the cloud comes with some limitations and disadvantages regarding security. Denial-of-service attacks (DoS/DDoS) are one of the major security challenges in emerging cloud computing environments. In this paper, the main objective is to propose a DoS/DDoS attack detection system for Cloud environments using the most popular CICIDS2017 benchmark dataset and applying multiple Machine Learning (ML) techniques by considering both the Wednesday and Friday afternoon traffic log files. The implementation results of our model based on the eXtreme Gradient Boosting (XGBoost) algorithm demonstrate its ability to detect intrusions with a detection accuracy of 99.11% and a false alarm rate of about 0.011%.

1 INTRODUCTION

Cloud computing is the subject of the era and is the current keen domain of interest to organizations. It is defined by the NIST (National Institute of Standards and Technology) as "A model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" (Mell and Grance, 2011).

Apart from the characteristics and services provided by cloud computing, there are several security issues that act as a hindrance to its adoption and may lead to catastrophic impacts on availability, confidentiality, and integrity.

In this paper, we are interested in the detection of one of the most important security problems facing cloud computing which is the Denial-of-Service attack (DoS/DDoS) that proved extremely damaging to the availability of the services.

Fundamentally, the DDoS attacks are accomplished through a network of remotely controlled computers or zombies via command and control (C&C) channels by sending overwhelming amounts of data that exceed the bandwidth capabilities of the targeted victim. Figure 1 shows the DDoS attack in the cloud environment.

There are three different types of DoS/DDoS attacks, such as volume-based which utilize many computers and internet connections to flood a website with traffic so that an overwhelming amount clogs up the website's available bandwidth. E.g., a UDP flood attack in which an attacker overwhelms random ports on the targeted host so that as more UDP packets are received and answered, the system is unable to handle the huge volume of requests and thus becomes unresponsive. Unlike volume-based attacks, protocol attacks aim to exhaust server resources rather than bandwidth. Attackers overwhelm websites and these server resources by making phony protocol requests to consume the available resources. E.g., Smurf DDoS. Attackers exploit Internet Control Message

^a <https://orcid.org/0000-0002-0934-3224>

^b <https://orcid.org/0000-0003-0551-4927>

^c <https://orcid.org/0000-0001-9250-7885>

Protocol (ICMP) packets that contain the victim’s spoofed IP and then broadcast the IP to a computer network using an IP broadcast address. If the number of devices on the network is large enough, the victim’s computer will be flooded with traffic since most devices on the network respond by default to the source IP address. Generally, application-layer attacks require fewer resources than volume-based attacks and protocol attacks. This type of attack looks to disrupt specific functions or features of a website. E.g., Slowloris, software created by Robert “Rsnake” Hansen (Shorey et al., 2018), that enables a single computer to take down a web server. It works by opening multiple connections to the targeted web server and keeping them open as long as possible. It constantly sends partial HTTP requests which are never completed. Immediately, the target server’s maximum connection pool is filled and further connection attempts are rejected.

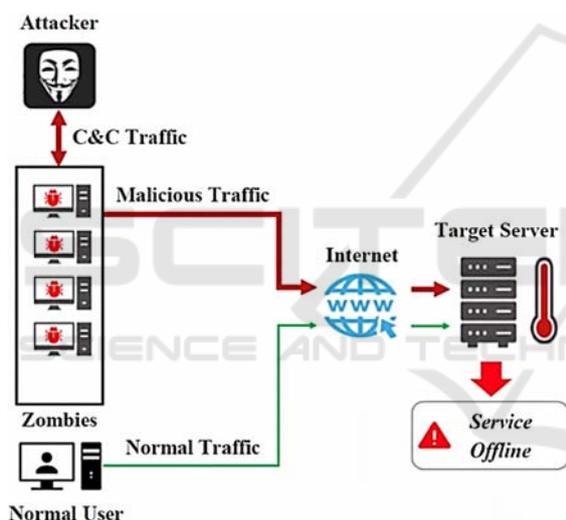


Figure 1: DDoS Attack Architecture.

These attacks have been around for several years. In early 2015, a heavy DDoS attack targeted Greatfire.org and cost it a high bill of \$30,000 a day on Amazon’s Elastic Computing Cloud (EC2) (Attaran et al., 2018). In 2016, a two-hour DDoS attack was carried out on Amazon, Twitter, and Spotify resulting in huge financial losses due to service interruptions (Gaurav et al., 2017). In recent years, the scale of these attacks has increased drastically. On 28 February 2018, Akamai reported a 1.3 TBps attack on GitHub. A few days later, Arbor Networks reported a 1.7 TBps attack (Shaar and Efe, 2018). According to NexuSGuard’s Q1 2020 Threat Report (DDoS Threat Report Q1, 2020), the DDoS attacks jumped more than 542% compared with the

last quarter of 2019 and more than 278% year-over-year and according to NETSCOUT’s ATLAS Security Engineering & Response Team (ASERT), in the first quarter of 2021, approximately 2.9 million DDoS attacks were launched by the threat actors, and it is a 31% increase from the same time in 2020.

Through the above statistics, we understand the necessity of an effective and early detection, mitigation, and prevention of DDoS attacks.

It is obvious that a Cloud IDS should analyze large volumes of network traffic data, detect efficiently the new attack behaviors, and reach high accuracy with low error rates. However, preprocessing, analyzing, and detecting intrusions in Cloud environments using traditional techniques have become very costly in terms of computation, time, and budget. Therefore, efficient DoS/DDoS attack detection in Cloud environments requires the adoption of new intelligent techniques such as Machine Learning (ML) and Deep Learning (DL).

Our objective in this paper is to implement Machine Learning (ML) classifiers to detect DoS/DDoS attacks with the aim to achieve fast detection rates, low error rates, and high accuracy with an affordable computational cost.

The main contributions of this paper can be summarized as follows:

- (i) Reviewing the significant related works to investigate all aspects of ML models’ contribution in the classification of network traffic traces in cloud environments as normal and DoS/DDoS.
- (ii) Proposing a powerful DoS/DDoS attack detection system based on ML techniques reaching high detection accuracy and low error rates.
- (iii) Adapting a procedure to select appropriate features for DoS/DDoS attack detection.
- (iv) Applying performance improvement strategies to reduce computation time and save processing power.
- (v) Our practical way of classifying the incoming traffic in this paper is advantageous in contrast to other comparable works as we provide a summary of the execution results of each detection model.

The remainder of this paper is structured as follows. A comprehensive review of some related works is provided by discussing the main contributions of the proposed solutions in section 2. Section 3 presents our experimentation with a discussion of the obtained results. Finally, section 4 states the conclusion with some future directions.

2 RELATED WORK

Several researchers have oriented their research axes to detect DoS/DDoS attacks using multiple methods and techniques. In this section, we summarize some of the recent works in the detection of DoS/DDoS attacks using different ML/DL approaches.

The focus of (Virupakshar et al., 2020) is on bandwidth and connection flooding types of DDoS attacks. The Decision Tree (DT), K-Nearest Neighbors (KNN), Naïve Bayes (NB), and Deep Neural Network (DNN) algorithms were used for the detection of DDoS attacks in the cloud environment. The DNN model has been chosen as it has the highest accuracy and precision values of about 96% using the dynamically generated dataset from the OpenStack-based private cloud platform. The main limitation of this paper is that it validates the proposed approach with an obsolete dataset namely KDDCUP99.

(Bhardwaj et al., 2020) propose a novel architecture that combines a stacked sparse Autoencoder (AE) for feature learning with a Deep Neural Network (DNN) for the classification of network traffic into DDoS and normal network traffic. A comparative analysis of the proposed approach has been conducted with ten state-of-the-art approaches and validated based on the CICIDS2017 and the NSL-KDD standard datasets. The proposed approach yields competitive results as compared to other state-of-the-art methods giving an accuracy of 98.43% over the NSL-KDD and 98.92% using the CICIDS2017. However, certain limitations in this work are evident and the most obvious one is the lack of information regarding the detection time of the proposed model.

(Wei et al., 2021) proposed a hybrid method namely AE-MLP to separate the DDoS attacks from the normal network traffic. The AE identifies the most significant features automatically and the MLP takes the selected features as input and classifies the DDoS attacks based on the attack types. The suggested technique was evaluated based on the CICDDoS2019 dataset. According to the obtained results, the precision, recall, and accuracy are measured as 97.91%, 98.48%, and 98.34%, respectively. One of the advantages of this work is its ability to detect different types of attacks. However, it requires high computational resources during the training phase of the proposed model.

(Azizan et al., 2021) present an analysis of IDS using three popular classification algorithms, which are random forest (RF), decision jungle (DJ), and support vector machine (SVM). The ML-based NIDSs are implemented and tested using the CIC-IDS2017. The obtained results show that the SVM

has the best overall results in detecting the DDoS attacks with an average accuracy of 98.18%, a precision of 98.74%, and an average recall of 95.63% and thus can be used as an IDS. This paper limited the classification process to only three ML algorithms which may be extended to explore more classifier systems.

The research proposed by (Kumar et al., 2022) identifies modern DDoS attacks based on the light gradient boosting method (LGBM) and the extreme gradient boosting (XGBoost) using the openly available dataset CICDDoS 2019. These two ML methods have been selected because of their superior prediction ability in high volumes of data in less computation time. According to the experimental results, the highest accuracy is obtained by the XGBoost-based model with an average of 94.80% in 229 seconds. A limitation of this work is that all the instances present in the dataset cannot be processed, even with the use of high-end machines.

3 EXPERIMENTATION AND DISCUSSION

In this section, we first give the performance metrics used to evaluate our model. Then, we examine the details of the CICIDS2017 dataset used for deployment and validation of our detection method, along with the data pre-processing procedure. Finally, we discuss the experimental results that we attained.

3.1 Performance Metrics

The ability of IDS to make the correct predictions considers the measure of its effectiveness. Depending on the comparisons between the results that are predicted via IDS and the true nature of the event, there are four prospect outputs that are illustrated in Table 1 well known as the confusion matrix. These four outcomes are:

- True Positives (TP): The cases in which the IDS predicted «Malicious» and the actual output was also « Malicious ».
- True Negatives (TN): The cases in which the IDS predicted «Benign» and the actual output was «Benign».
- False Positives (FP): The cases in which the IDS predicted «Malicious» and the actual output was « Benign ».
- False Negatives (FN): The cases in which the IDS predicted «Benign» and the actual output was « Malicious ».

For an IDS to be effective, the FP and FN rates should be minimized, and TP and TN rates should be maximized.

Table 1: Confusion matrix.

Actual class	Predicted class	
	Malicious	Benign
Malicious	True Positive (TP)	False Negative (FN)
Benign	False Positive (FP)	True Negative (TN)

These performance metrics are not dependent on the size of the training and testing samples and can be helpful in assessing the performance of the model:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (1)$$

$$\text{Precision} = \frac{TP}{TP + FP} \quad (2)$$

$$\text{Recall} = \frac{TP}{TP + FN} \quad (3)$$

$$\text{F1 - Score} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}} \quad (4)$$

$$\text{False Alarm Rate} = \frac{FP}{FP + TN} \quad (5)$$

3.1.1 Experimentation Based on the CICIDS2017 Dataset

To successfully build an efficient ML-based DoS/DDoS attack detection system, a reliable up-to-date labeled dataset is required.

There exist a number of such datasets that have been used by researchers to evaluate the performance of their proposed intrusion detection and prevention approaches. Examples of these datasets are: KDDCUP99, NSL-KDD, CICIDS2017, and CICDDoS2019, etc.

Based on our study of the well known available datasets since 1998, many such datasets are obsolete and unreliable while others lack feature sets, traffic diversity or do not cover a wide range of attacks which cannot reflect the current trends.

In this paper, we used the CICIDS2017 dataset generated by the Canadian Institute of Cybersecurity as it satisfies all the criterias of a reliable benchmark dataset (Sharafaldin et al., 2018).

The CICIDS2017 dataset contains an abstract normal behavior of twenty-five users based on the HTTP, HTTPS, FTP, SSH, and email protocols with

several different attack traces, including Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltration, Botnet, and DDoS. this dataset is spanned over eight different files available in a comma-separated values (CSV) format. The total rows contained in those eight files are 2,830,743 rows; each row has seventy-eight features and is labeled as Benign or one of fourteen types of attacks. The details of all those files are shown in Table II. Later, Table 2 lists the features within the CICIDS2017 dataset.

Table 2: The Details of files containing the CICIDS2017 dataset.

File's Name	Normal Flows	Attack Flows	Class Labels
Monday-WorkingHours.pcap_ISCX.csv	529,918	0	Benign
Tuesday-WorkingHours.pcap_ISCX.csv	432,074	13,835	Benign, FTP-Patator, SSH-Patator
Wednesday-WorkingHours.pcap_ISCX.csv	432,074	252,672	Benign, DoS GoldenEye, DoS Hulk, DoS Slowhttptest, DoS slowloris, Heartbleed
Thursday-WorkingHours-Morning-WebAttacks.pcap_ISCX.csv	168,186	290,782	Benign, Web attack-Brute Force, Web Attack-Sql Injection, Web Attack-XSS
Thursday-WorkingHours-Afternoon-Infiltration.pcap_ISCX.csv	288,566	36	Benign, Infiltration
Friday-WorkingHours-Morning.pcap_ISCX.csv	189,067	1966	Benign, Bot
Friday-WorkingHours-Afternoon-PortScan.pcap_ISCX.csv	183,910	41,835	Benign, PortScan
Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv	127,537	158,930	Benign, DDoS

Table 3: Features of the CICIDS 2017 dataset.

N.	Feature	N.	Feature
1	Destination Port	40	Max Packet Length
2	Flow Duration	41	Packet Length Mean
3	Total Fwd Packets	42	Packet Length Std
4	Total Backward Packets	43	Packet Length Variance
5	Total Length of Fwd Packets	44	FIN Flag Count
6	Total Length of Bwd Packets	45	SYN Flag Count
7	Fwd Packet Length Max	46	RST Flag Count
8	Fwd Packet Length Min	47	PSH Flag Count
9	Fwd Packet Length Mean	48	ACK Flag Count
10	Fwd Packet Length Std	49	URG Flag Count
11	Bwd Packet Length Max	50	CWE Flag Count
12	Bwd Packet Length Min	51	ECE Flag Count
13	Bwd Packet Length Mean	52	Down/Up Ratio
14	Bwd Packet Length Std	53	Average Packet Size
15	Flow Bytes/s	54	Avg Fwd Segment Size
16	Flow Packets/s	55	Avg Bwd Segment Size
17	Flow IAT Mean	56	Fwd Header Length
18	Flow IAT Std	57	Fwd Avg Bytes/Bulk
19	Flow IAT Max	58	Fwd Avg Packets/Bulk
20	Flow IAT Min	59	Fwd Avg Bulk Rate
21	Fwd IAT Total	60	Bwd Avg Bytes/Bulk
22	Fwd IAT Mean	61	Bwd Avg Packets/Bulk
23	Fwd IAT Std	62	Bwd Avg Bulk Rate
24	Fwd IAT Max	63	Subflow Fwd Packets
25	Fwd IAT Min	64	Subflow Fwd Bytes
26	Bwd IAT Total	65	Subflow Bwd Packets
27	Bwd IAT Mean	66	Subflow Bwd Bytes
28	Bwd IAT Std	67	Init Win bytes forward
29	Bwd IAT Max	68	Init Win bytes backward
30	Bwd IAT Min	69	act data pkt fwd
31	Fwd PSH Flags	70	min_seg_size forward
32	Bwd PSH Flags	71	Active Mean
33	Fwd URG Flags	72	Active Std
34	Bwd URG Flags	73	Active Max
35	Fwd Header Length	74	Active Min
36	Bwd Header Length	75	Idle MeanIdle Std
37	Fwd Packets/s	76	Idle Max
38	Bwd Packets/s	77	Idle Min
39	Min Packet Length	78	Label

The methodology that has been carried out in this work is depicted in Figure 2.

Specifically, the proposed method consists of four stages: a) Data Preprocessing, b) Feature Selection and Extraction, c) Classification, d) Classification Results. We detail in the following sub-sections the functionalities of each used bloc/phase in our approach.

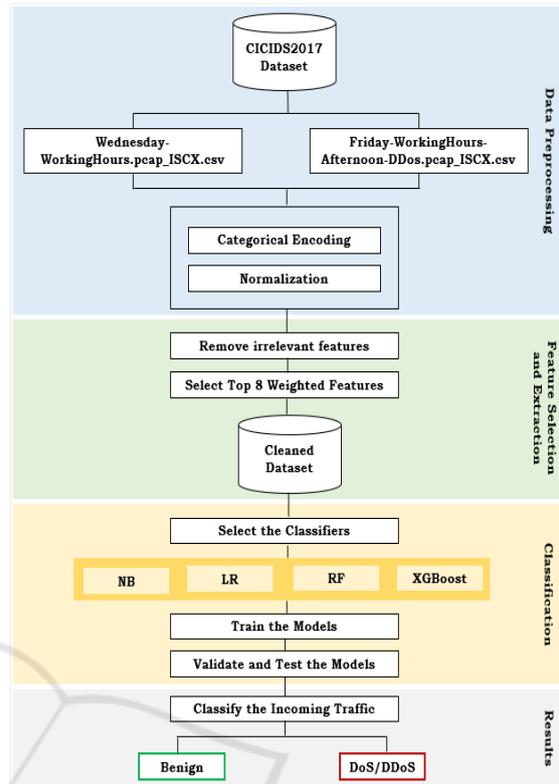


Figure 2: Our research methodology for DoS/DDoS attacks detection.

We should note that our experiments have been conducted using a Windows 10 – 64 bits PC with 16 GB RAM and CPU Intel(R) Core-i7 11370H.

a) Data Preprocessing

In this study, we used both the "Wednesday-WorkingHours.pcap_ISCX.csv" and the "Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv" combined in the same CSV file. Figure 3 depicts the frequency distribution of each class label within the obtained dataset.

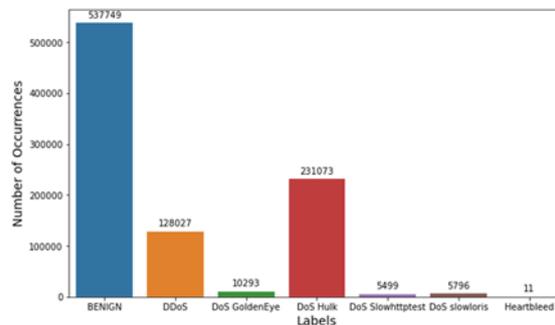


Figure 3: Frequency distribution of class labels within the obtained dataset.

As the class labels are not balanced, we have mixed the different attack traces together in one class label named "DoS/DDoS" and we obtained the result shown in Figure 4.

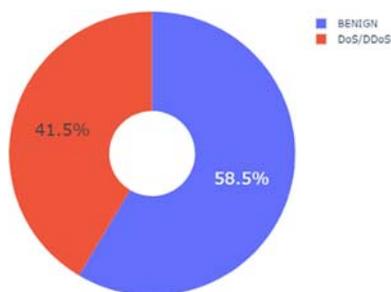


Figure 4: Frequency distribution of class labels.

To make the records in both training and testing subsets extracted from this dataset ready for processing by our proposed IDS, we handle the data preprocessing using the two following operations:

- **Categorical Encoding:** Our dataset contains only one non-numeric feature namely "Label" corresponding to "BENIGN" and "DoS/DDoS". As this study does not distinguish between different categories of attacks, the attributes "BENIGN" are converted to "0". Whereas, the attributes corresponding to the "DoS/DDoS" are converted to "1".
- **Normalization:** We used the process of normalization known as the statistical Z-score method as it helps reduce classification errors significantly and allows the model to converge faster. The Statistical normalization (Z-score) formula is as follows:

$$X = \frac{x - \mu}{\alpha} \tag{6}$$

The value x of a feature A is transformed in X according to formula (6). μ is the mean and α is the standard deviation of a given attribute.

b) Feature Selection and Extraction

In this phase, we removed the features having the same values for all the rows, namely "Bwd PSH Flags", "Bwd URG Flags", "Fwd Avg Bytes/Bulk", "Fwd Avg Packets/Bulk", "Fwd Avg Bulk Rate", "Bwd Avg Bytes/Bulk", "Bwd Avg Packets/Bulk", and "Bwd Avg Bulk Rate" because their presence will consume more overhead for loading and processing. We also eliminated all the rows whose features have values equal to "Infinity" or "nan" because ML classifiers cannot utilize them either in training or in testing phases.

After removing the irrelevant attributes, we perform feature selection with feature importance based on the Extra Tree classifier to highlight the most important or relevant features to the output variable. The top eight features ranked and selected for the prediction of our target variable are depicted in Figure 5 and described in Table 4.

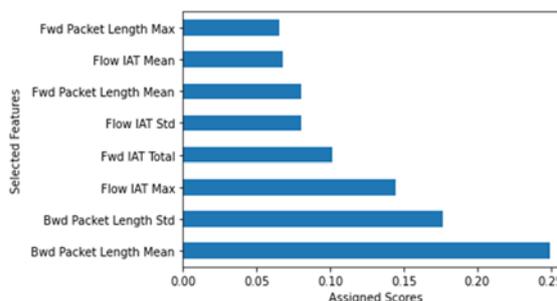


Figure 5: Feature Importance scores.

Table 4: Description of Features used in our IDS.

Feature	Description
Fwd Packet Length Max	Maximum size of the packet in the forward direction of the data flow.
Flow IAT Mean	Mean time between two packets sent in the forward direction of the data flow.
Fwd Packet Length Mean	Mean size of the packet in the forward direction of the data flow
Flow IAT Std	Standard deviation time between two packets sent in the data flow
Fwd IAT Total	Total time between two packets sent in the forward direction of the data flow.
Flow IAT Max	Maximum time between two packets sent in the forward direction of the data flow.
Bwd Packet Length Std	Standard deviation size of the packet in the backward direction of the data flow.
Bwd Packet Length Mean	Mean size of the packet in the backward direction of the data flow.

c) Classification

In this phase, we randomly divided our dataset into 70% for training and 30% for testing with the Naïve Bayes (NB), Logistic Regression (LR), Random Forest (RF), and eXtreme Gradient Boosting (XGBoost) algorithms that we chose based on multiple factors:

- Knowledge of data's structure and complexity;
- Processing speed of the classification task;
- Features taken into account when training the model for the best possible outcome and accuracy;
- Parameters such as the number of iterations directly relate to the training time needed when generating the output.

A short description of these ML classification algorithms is presented in Table 5.

Table 5: Description of the four classification algorithms.

Classifier	Description
NB	<p>A probabilistic ML algorithm based on the Bayes Theorem which is a way of finding a probability when certain other probabilities are known by considering the following formula:</p> $P(H E) = \frac{P(E H) * P(H)}{P(E)} \quad (7)$ <p>Where $P(H E)$ indicates the posterior Probability of the hypothesis given that the evidence is true, $P(E H)$ indicates the likelihood of the evidence given that the hypothesis is true, $P(H)$ is the prior probability of the hypothesis, and $P(E)$ is the prior probability that the evidence is true.</p>
LR	<p>The logistic regression algorithm is used to solve classification problems. The model is defined as follows:</p> $P(Y = 1 x) = \frac{e^{wx+b}}{1 + e^{wx+b}} \quad (8)$ $P(Y = 0 x) = \frac{e^{-wx-b}}{1 + e^{-wx-b}} \quad (9)$ <p>Where w indicates the weight, b indicates the bias, and $wx+b$ is regarded as the linear function of x. Compare the preceding two probability values. The class with a higher probability value is the class of x.</p>
RF	<p>An ensemble ML paradigm that independently builds several decision trees and merges them together to make predictions more accurate and stable. Decision Tree is a binary or non-binary tree structure, where each non-leaf node denotes a test on a feature attribute, each branch represents the output of a feature attribute in a certain value range, and each leaf node holds a class label.</p>
XGBoost	<p>A Decision Tree-based ensemble learning algorithm that uses Gradient Descent as the underlying objective function and comes with a lot of flexibility while predicting the desired results by consuming the computational power in an optimal way.</p>

d) Classification Results

In this section, we report all the obtained results. The different diagrams corresponding to the confusion matrix of the proposed ML models are shown in Figure 6.

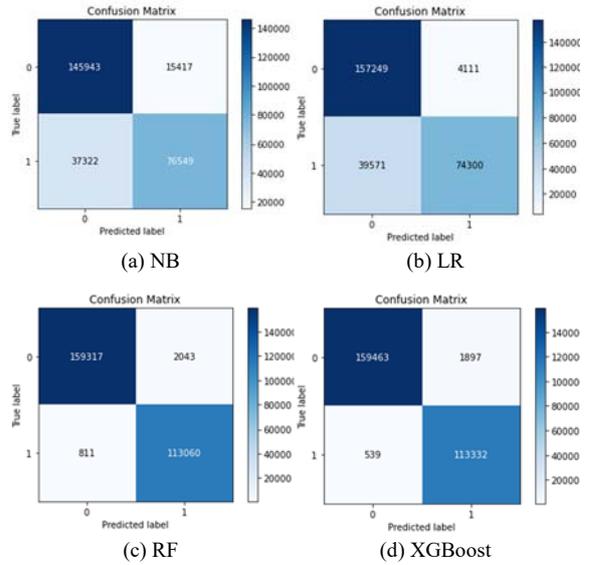


Figure 6: Diagrams of confusion matrix of the proposed ML Models.

The 'x-axis' of each confusion matrix presents the predicted class label and the 'y-axis' presents the true label. By comparing the number of records in each class label, we can observe that the records in each class are not uniformly distributed. Hence, we analyze the performance of the selected ML models using different classification parameters as detailed in Table 6.

According to Table 6, we remark that all the proposed models provide promising classification results. Thus, all are valid and acceptable models.

From the perspective of the calculation time, NB is the best classification model as it spends only 0.03 minutes. However, it provides the lowest detection accuracy of 80.84% and the highest false error rate of 0.095%.

For this dataset, the XGBoost is likely to be the best classification model as it marginally outperforms the other models with its highest Accuracy of 99.11%, Precision of 99.12%, Recall of 99.11%, F1-Score of 99.12%, and False Alarm Rate of approximately 0.011%.

Table 6: Overall prediction performance of the ML models.

Model Name	Accuracy	Precision	Recall	F1-Score	False Alarm Rate	Training Time
NB	80.84%	81.12%	80.84%	80.43%	0.095%	0.03 min
LR	84.13%	86.04%	84.13%	83.45%	0.025%	0.43 min
RF	98.96%	98.97%	98.96%	98.96%	0.012%	8.48 min
XGBoost	99.11%	99.12%	99.11%	99.12%	0.011%	4.43 min

To illustrate the diagnostic ability of our binary classifier systems, we create the receiver operating characteristic curves (ROC) by plotting the true-positive rates against the false-positive rates at various threshold settings. Higher the Area Under the Curve (AUC) measure, the better the model is at predicting 0s as 0s and 1s as 1s. Figure 7 presents the ROC curves with their corresponding AUC scores for our proposed classification models.

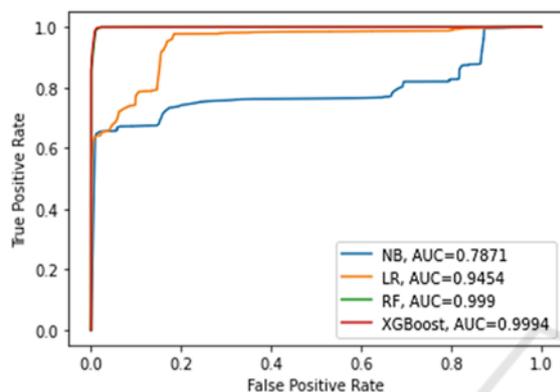


Figure 7: ROC Curves of the proposed ML models.

It is clearly observable from the obtained results that the highest AUC score is obtained with the XGBoost classifier with an AUC value of 99.94%. Therefore, it is the most appropriate model to distinguish between Dos/DDoS and benign network traffic features.

To validate the obtained results, we compare our proposed approach with some state-of-the-art DoS/DDoS detection methods based on the used ML/DL techniques and the highest detection accuracy. The comparison results are summarized in Figure 8.

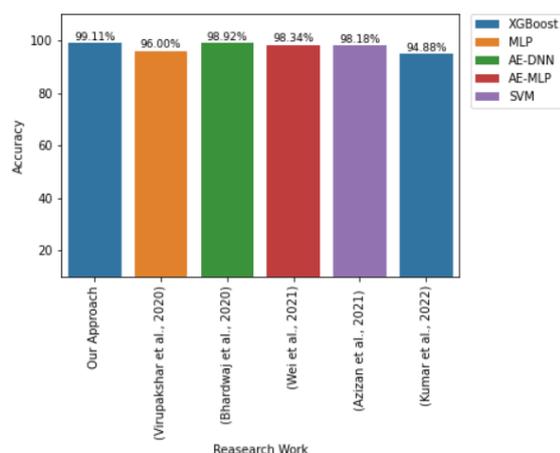


Figure 8: Comparison of the proposed approach with research approaches of related works.

We remark from Figure 8 that our proposed approach achieves the best accuracy reaching an average of 99.11% with the XGBoost classification algorithm.

Compared to the literature, the results obtained by the proposed approach are very satisfactory.

4 CONCLUSION

Detection of DoS/DDoS attacks in a cloud environment is a challenging task because they are more dangerous than other possible attacks as they are difficult to detect, easy to carry out, and rather difficult to predict the target of the attack. In this paper we propose a ML-based DoS/DDoS attacks detection system for cloud environments by considering the CICIDS2017 dataset.

Our experimental results demonstrate that our proposed IDS outperforms several recent works in terms of detection accuracy and error classification rates.

The results obtained in this paper are very interesting. However, there are still several contributions that could be furthered. In the future, we intend to extend this work to include newer ML techniques, with the aim of improving the performance against a wider range of cloud attacks.

REFERENCES

Mell, P. & Grance, T. (2011), The NIST Definition of Cloud Computing, Special Publication (NIST SP), National Institute of Standards and Technology, Gaithersburg, MD, [online], <https://doi.org/10.6028/NIST.SP.800-145>. Last checked on March 22, 2022.

Shorey, T., Subbaiah, D., Goyal, A., Sakxena, A., & Mishra, A.K. (2018). Performance Comparison and Analysis of Slowloris, GoldenEye and Xerxes DDoS Attack Tools. *2018 International Conference on Advances in Computing, Communications, and Informatics (ICACCI)*, 318-322.

Attaran, Mohsen & Woods, Jeremy. (2018). Cloud computing technology: improving small business performance using the Internet. *Journal of Small Business & Entrepreneurship*. 13. 94-106. 10.1080/08276331.2018.1466850.

Gaurav Somani, Manoj Singh Gaur, Dheeraj Sanghi, Mauro Conti, Rajkumar Buyya, DDoS attacks in cloud computing: Issues, taxonomy, and future directions, *Computer Communications*, Volume 107, 2017, Pages 30-48, ISSN 0140-3664, <https://doi.org/10.1016/j.comcom.2017.03.010>.

- F, International & Shaar, Fadi & Efe, Ahmet. (2018). DDoS Attacks and Impacts on Various Cloud Computing Components.
- DDoS Threat Report 2020 Q1. 2020. Nexusguard. <https://blog.nexusguard.com/threat-report/ddos-threat-report-2020-q1>, August 23, 2021.
- Virupakshar KB, Asundi M, Channal K, Shettar P, Patil S, Narayan DG (2020) Distributed Denial of Service (DDoS) attacks detection system for OpenStack-based Private Cloud. *Procedia Computer Science* 167:2297–2307.
- Bhardwaj, Aanshi & Mangat, Veenu & Vig, Renu. (2020). Hyperband Tuned Deep Neural Network With Well Posed Stacked Sparse AutoEncoder for Detection of DDoS Attacks in Cloud. *IEEE Access*. 8. 181916-181929. 10.1109/ACCESS.2020.3028690.
- Yuanyuan Wei, Julian Jang-Jaccard, Fariza Sabrina, Amardeep Singh, Wen Xu, Seyit Camtepe, "AE-MLP: A Hybrid Deep Learning Approach for DDoS Detection and Classification", *IEEE Access*, vol.9, pp.146810-146821, 2021.
- Azizan, Adnan & Mostafa, Salama & Mustapha, Aida & Mohd Foozy, Cik Feresa & Abd Wahab, Mohd Helmy & Mohammed, Mazin & Khalaf, Bashar. (2021). A Machine Learning Approach for Improving the Performance of Network Intrusion Detection Systems. *Annals of Emerging Technologies in Computing*. 5. 201-208. 10.33166/AETiC.2021.05.025.
- Kumar, V & Kumar, A & Garg, S & Payyavula, S. (2022). Boosting Algorithms to Identify Distributed Denial-of-Service Attacks. *Journal of Physics: Conference Series*. 2312. 012082. 10.1088/1742-6596/2312/1/012082.
- Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani, "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", *4th International Conference on Information Systems Security and Privacy (ICISSP)*, Portugal, January 2018.