DDoS Attacks on Cloud: A Combination of Machine Learning Approaches

Hassan Essam
Electrical and Computer Engineering
University of Ottawa
Ottawa, Canada
hahme107@uottawa.ca

Ashraqat Torky
Electrical and Computer Engineering
University of Ottawa
Ottawa, Canada
atork095@uottawa.ca

Yomna Abdelsattar
Electrical and Computer Engineering
University of Ottawa
Ottawa, Canada
yabde005@uottawa.ca

Tokka Hassan
Electrical and Computer Engineering
University of Ottawa
Ottawa, Canada
thass019@uottawa.ca

***Abstract*—Cloud computing is a revolution in IT technology that provides scalable and virtualized resources that can be used on demand where it offers high flexibility, zero maintenance and low infrastructure cost. These cloud-based services use some networking protocols that contain bugs and vulnerabilities, which open the door for intrusions done by attackers. DDoS (Distributed Denial of Service) attacks are one of the most used attacking techniques. These types of attacks continue to develop at a rapid pace so detecting and overcoming them have become more challenging. In this paper we will review some related work in this area, and provide a new methodology to tackle that type of attack detection. The evaluation shows that the new methodology gets better results compared to a baseline model -random forest- where it scores 94% accuracy and F1-score while the baseline scores 93% accuracy and F1-score.**

I. Introduction

Distributed computing which is also known as cloud computing is an extraordinary revamping for centralism of various PC administrations under one server or virtualization over many servers. Users and institutions are removing their programs, data, and backups from PCs and desktop computers to safely store them in the cloud. Distributed computing is an exceptionally solid contender inside the IT world and it offers "pay as you go" with minimal expense administrations.
Generally, huge enterprises and associations have moved their data and work into the cloud. This modern approach helps greatly with issues regarding time, administration and set-up effort, and much more, by offering types of services exactly at the level of effort you can provide to save time, money, and quality. That's why cloud computing offers various types of services but the top services which are being utilized regularly are Services as a service (SaaS), Platform as a service (PaaS), and Infrastructure as a service (IaaS). However, as this industry is developing and growing, many challenges and risks appear with it in sight.

Given the fact that there's a massive amount of data and organizational work present on the cloud, it became a very attractive victim to cyber-attacks. The more the cloud grows, the more the attacks and the risks develop. Security is an essential concern in this matter. In the paper [8], the authors use a metaphor of banks to simply illustrate the security concern in the cloud. If you have entrusted a band with your jewels, but then this bank turns out to have no safety officers or security guards. In this case, no matter how you need to store your jewels somewhere, you will not risk it. In the cloud world, data is like jewels, it represents the value and the core of many organizations. Hackers are like bank robbers, they wouldn't miss a server without a profound security system. The types of attacks that can cause serious harm to the data or the models present on the cloud are evolving and increasing. During our research we found many authors agreeing on the fact that DDoS Attack is an alert for cloud users. That is on the grounds that cloud services are typically conveyed by HTTP convention, which is defenseless against HTTP attacks. HTTP DDoS attacks disturb the servers and services by sending surges of messages and malformed packets. Millions of these malformed packets or fake messages are sent from various distributed or virtually distributed systems to one single server. This results in slowing down all the services on the targeted server and disabling it from responding to real messages.
DDos attacks' target could be a network or an application. Huge organizations, national and global service providers as well as governmental companies being the main target of DDoS attacks, need to develop their security measures as the attacks develop day by day. The traditional methods are no longer sufficient. If we were to address the problem of DDoS detection from an artificial intelligence point of view, it would mainly be a classification problem in machine learning with considerable concern in the prediction speed.

DoS Attacks are mainly planned to prevent real clients and users from getting to a particular organization's service or network resources. The OSI model (Open Systems Interconnection), is key in understanding the kinds of DDoS attacks

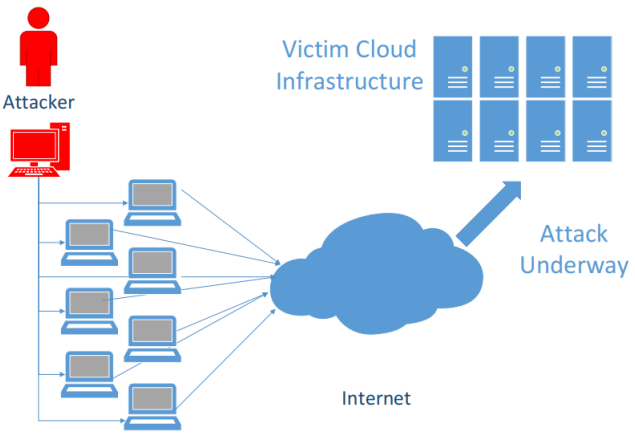

Fig. 1. Typical Architecture of DDoS Attacks

out there. DDoS attacks target explicit layers of a network (application layer, protocol layer, and others). Knowing the objective of the attacker which motivates them to do the attack, may be of assistance to us to help us detect the attacks. Some motivations could be financial or economic gain. For example, disappointed or frustrated employees may perform attacks against their corporations as means of objection. Also, given that they already know a lot of confidential information about the company's system, it makes the mission easier for them.

Another objective could be an intellectual challenge. Young attackers tend to show off their ability through a real-life hacking experiment to also enhance their methodologies. On another note, if the attack is on a country's governmental level, the goal could be cyber warfare. A country with political problems would launch the attack against its rivals to put them in tough situations. If we narrow the scope a little bit from the governmental level, we will find the motivation of different ideological beliefs. Some DDoS attackers who have strong religious or political beliefs may be highly motivated to perform the attacks. An example of this is what happened in 2007 in Estonia, Iran of sabotage attack [8].

A. Types of DDos Attacks According to [8]

1) SYN Flood attack: SYN flood attack breakdown an irregularity in TCP connection that incorporates three-way handshake, regularly sync packets are sent towards the other host to check whether or not the host side is inactive, accordingly, it offers back its ACK pulse in addition to with SYN packet. Eventually, another connection can be established. However, in the occurrence of SYN flood, the sender sends numerous SYN flood requests and does not react back to the host ACK, at the same time it keeps on sending demands. The host side keeps on hanging waiting for the ACK for each message and ties many other resources for new connections yet all in vain and in the long run it brings about Denial of Service.

2) Ping of death: This attack incorporates sending malformed rings to systems. Generally, the size of an IP packet is 65,535 bytes and the data link layer handles a size of a maximum of 1500 bytes. This results in a huge IP packet dividing into various smaller multiple packets which are for the most part known as fragments and afterwards, the target side reassembles them into the original complete packet. On account of the ping of death, the receiver gets a ping of in excess of 65,535 packets which are much bigger than expected and this can lead to memory flood, causing Denial of service like what happened in Neupane, Neely, 2019 [8].

3) NTP amplification: To overwhelm the designated side with UPD traffic, the network time protocol takes advantage of its prepress exploits. It includes the technique query to response framework and an explicit proportion is being declared as 1:20 to 1:200. On the off chance that any attacker gets that list of open NTP servers by utilizing any tools. For example Metasploit can hit the desired target then dispatch a DDoS attack (Hong, Nhlabatsi, 2019) [8].

4) HTTP flood: In order to attack web servers or online applications, the attackers exploit GET or POST HTTP requests. If one http request turns out trying to allocate the maximum amount of resources, this is when the attack is most effective.

5) UDP flood: UDP is defined as an attack which targets the server end. It uses the user datagram protocol packets to work. As usual, the attack randomly selects some ports on distant hosts and targets them with the flood. As a result, the targeted host keeps checking the applications on the ports. If it doesn't find applications listening, it replies with ICMP wich can eventually lead to inaccessibility.

6) Zero-day attacks: Zero-day attacks: it encapsulates all-new attacks which may also include all anonymous attacks. It exploits all the vulnerabilities for attackers. The term Zero-day attack is popular within the hackers' community.

II. State of The Art Review

A. Related Work

In this part, We'll go through some of the well-known collection of literatures about DDOS attack detection approaches.

Simple techniques were developed to overcome DDoS attacks like the CAPTCHA puzzle but recent studies show that it is not effective. [1] Other techniques as IP address monitoring and IP traceback defense are widely used. [2] [3]

Dantas Y et al. [4] developed a technique that works with network layer DDoS attacks but fails with the application layer as the technique is based on the concept of the notion of a state and the DDoS attacks in the application layer do not have a notion of the state.

Several machine learning and data mining techniques have been recently used to detect and overcome DDoS attacks. Alkasassbeh et al. [5] developed a technique that uses 27 features and machine learning techniques like Multilayer Perceptron, Random Forest, and Naïve Bayes. This methodology uses a DDoS attacks database collected at different network layers like SIDDoS and HTTP flood.

Carl Livadas et al [6] used machine learning algorithms Naïve Bayes, J48, and Bayesian network classifiers to distinguish the traffic at command and control center into IRC traffic and real IRC traffic. Naïve Bayes outperformed the other two approaches, but J48 performs better for real IRC and non-IRC flows.

In this paper [7] they generate the DDoS attack in a secure environment. During the execution of a DDoS attack, traffic is generated at sink nodes and during this process, both the normal and suspicious traffic is recorded. Then an Intrusion Detection System SNORT was used which was given an input file from the server. It uses a rule-based tool but with changed rules not the default ones in order to detect all the DDoS attacks.

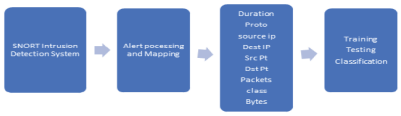

Fig. 2. SNORT Process of Attack Detection

In this paper [9] the authors investigated ML methodologies' ability through controlled attack scenarios for tasks such as the DDoS attack types' detection and identification like SYN Flood and GET Flood in different ratios with true users. They tried out different methodologies, specifically KNN and Random Forest machine learning algorithms. They trained, tested, and evaluated the models using precision, recall, accuracy, and F1-score given that many of the DDoS attacks data could be imbalanced so it made sense to use these metrics. Moreover, to give initial results the authors tested detecting the ping flood without training to get an idea of the generalization of the proposals. The authors also neglected network traffic information and stuck solely to using cloud telemetry data. There's also a big chance that by observing the utilized resources percentages, we could find traces to help us detect DDoS attacks in cloud environments, given that any DDoS attack usually aims to over-utilize resources. The paper's goal is not only to detect the occurrence of an attack, they also worked on identifying the type of DDoS attack. Of course, machine learning algorithms are very suitable to solve detection and identification problems. The inputs to the algorithms will be metrics natives generated by cloud telemetry services. In the figure, the cloud telemetry service gets the data from the computing resources of virtual machines. The monitoring of virtual machine resources is gathered and kept in the native telemetry database. This database creates datasets for the metrics which the VMs (virtual machines) collected as shown in the figure.

Afterward, the machine learning algorithm ingests the datasets to produce the trained models. Eventually, after the process of data collected, preparation, feature selection, and splitting, then training and testing, the evaluation of different machine learning methodologies begins. They evaluated various classifiers including K-Nearest Neighbor (kNN), Naive

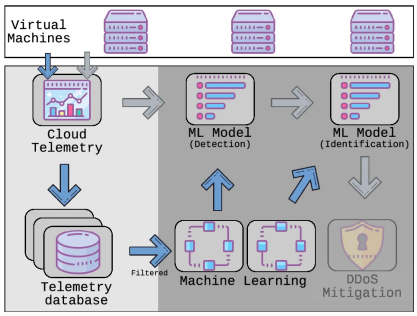

Fig. 3. Overview of the architectural components of the proposed solution

Bayes, Support Vector Machine (SVM), Decision Tree, Random Forest, and last but not least Multi-Layer Perceptron. The authors reached a champion model by evaluating the results with the metrics we mentioned (Precision, Recall, and F1-score). kNN achieved the best performance followed by Random Forest. For this reason, they focused on the results of these two classifiers.

Regarding the data labeling, the DDoS attacks were regarded as positive cases, while the absence of them or legitimate clients were regarded as negative cases. But that was in the detection phase since it was a binary classification problem. On another note, in the identification phase, which is also known as the multi-classification phase they had SYN Flood, GET Flood, and legitimate clients. So the records of the class of concern are considered positive cases and the rest of the records are negative cases. They computed an individual metric for each class. After that, they used a weighted average metric of all indicators while taking into consideration the number of records belonging to each category. In the training phase, when training with kNN at the beginning they performed hyperparameter tuning through varying the k parameter value between 1 and 15 trying out only odd numbers. They eventually declared the k parameter with 3 and used Euclidean distance to compute the algorithm steps. Similarly, for the random forest, they tried out different numbers of trees between 80 and 120 and eventually they used 100 trees because it achieved the best performance.

In the figure, the Precision, Recall, Accuracy, and F1-Score metrics of kNN and Random Forest are shown during the detection phase. The algorithms achieved high accuracy scores (above 87%). Both kNN and random forest achieved Recall Scores above 87% which means that the probability of correctly detecting the attacks is high. The precision metric's results were also promising with above 86% scores for both classifiers. This means that the legitimate clients' detection probability is quite good.

Algorithms	Accuracy (%)	Recall (%)	Precision (%)	F1-Score (%)
KNN	87.37	87.37	86.18	85.91
RF	87.28	87.28	86.16	85.51

Fig. 4. Results of ML algorithms in the detection phase

According to paper [8] the authors tried out various method-

ologies. Random forest and naive bayes are being compared because of their great results. In their binary classification experiments, Naive Bayes was the champion model. They used it to detect DDoS attacks in the application layer. They trained and analyzed the data using this classifier furthermore applied cross-validation of 60 folds for more solid results. They considered the high value of false-negative predictions as an alarm for network users (similar to the previous paper).

TP rate	FP rate	Recall	ROC area	PRC area	Class
0.000	0.06	0.000	0.055	0.01	192.168.8.1
1.000	0.14	1.000	1.00	1.00	10.0.2.15
0.985	0.10	0.985	1.00	1.00	89.31.143.1
Avg = 0.982	0.013	0.982	0.988	0.98	

Fig. 5. Results of naïve Bayes

In another paper's [10] implementation the system extracts network traffic features from network traffic samples and puts them in a buffer till it reaches a certain extent after that they are passed to a classifier which classifies that data into either normal or malicious. If malicious data was detected an alerting notification is passed through a cloud platform by traffic control protection systems. They used CIC-DoS, CICIDS2017, and CSE-CIC-IDS2018 and customized datasets as they include modern threats and DoS techniques.

Traffic sampling: The sFlow monitoring system is used to monitor the network of high-speed switches and routed devices continuously. But due to the high computation resulting from processing all the packets in the network the paper used the sampling technique. They used the sFlow protocol which uses n-out-of-N sampling technique. This technique simply picks randomly n numbers from the range of 1 to N, which is the number of packets. The n picked numbers are used to pick the position of packets, where the sample size is fixed.

Feature Extraction: The features were extracted from the headers of the network and transport layer packets of the TCP/IP architecture as it saves computation resources and makes the implementation in the internet service provider networks much easier. Five variables were selected as the most important ones out of the whole feature variables. They are source and destination ports, transport layer protocol, IP packet size, and TCP flags. Based on those, 33 variables were derived by using statistical measures that emphasize data variability.

Feature and Machine learning Algorithm selection: For the feature selection, two stages were done. The first was performing Recursive Feature Elimination with Cross-Validation (RFECV) with different numbers of features using different machine learning algorithms like Random Forest (RF), decision tree (DTree), logistic regression (LR), stochastic gradient descent (SGD), perceptron, and AdaBoost. Each model got the highest accuracies using a different set of features like shown in the table below:

#	MLA	No. of features	Accuracy
1	RF	28	0.996010
2	DTree	25	0.994182
3	LR	26	0.972327
4	SGD	16	0.969474
5	Perceptron	28	0.937256
6	AdaBoost	7	0.931131

Fig. 6. 10-fold RFECV results

In the second stage, another feature selection method was used using a proposed algorithm (Algorithm 1 [10]) in which random forest model was used. The output set of features from the second stage was tested on different models where Random Forest model got the highest accuracy and low false alarm rate which is essential in DDOS detection.

Input: database descriptors, variable importance threshold, accuracy threshold, and number of rounds
Output: selected variables
(1) **begin**
(2) *Create empty optimized model set;*
(3) **for** *i* ⟵ 1 **to** *Number of rounds* **do**
(4) *Define all the descriptor database variables as the current variables;*
(5) **while** *True* **do**
(6) *Split dataset in training and test partitions;*
(7) *Create and train the model using training data partition;*
(8) *Select the most important variables from the trained model;*
(9) *Calculate the cumulative importance of variables from the trained model;*
(10) **if** max (*cumulative importance of variables*) < *Variable importance threshold* **then**
(11) *Exit loop;*
(12) **end**
(13) *Train the model using only the most important variables;*
(14) *Test the trained model and calculate the accuracy;*
(15) **if** *Calculated accuracy* < *Accuracy threshold* **then**
(16) *Exit loop;*
(17) **end**
(18) *Add current model to optimized model set;*
(19) *Define the most important variables from the trained model as the current variables;*
(20) **end**
(21) **end**
(22) *Group the models by number of variables;*
(23) *Remove outliers from the grouped model set;*
(24) *Select the group of models with the highest frequency and their number of variables "N";*
(25) *Rank the variables by the mean of the importance calculated in step 7;*
(26) *Return the "N" most important variables;*
(27) **end**

Fig. 7. Algorithm 1

Results: They were evaluated using Precision (PREC), Recall (REC), and F-Measure (F1), the detection rate (DR), and false alarm rate (FAR) metrics. Where DR is the ratio between the number of attacks detected by the system and the actual number of attacks performed. FAR is the ratio between FP and the sum of FP and TN. Shown below is a table illustrating the performance of each dataset using the stated metrics [10]:

Dataset	DR	FAR	PREC	F1
CIC-DoS	0.936	0.0004	0.999	0.999
CICIDS2017	0.800	0.002	0.992	0.992
CSE-CIC-IDS2018	1.000	0.000	1.000	1.000
Customized	0.965	0.002	0.995	0.995

Fig. 8. Evaluation of system performance for all data sets.

The best performance was found in the CSE-CIC-IDS2018 dataset, which had dominant malicious traffic relative to normal traffic.

[11] proposed a detection algorithm using SVM classifier which searches for the pattern of ”bots” while they are visiting the websites

[12] proposed an anomaly detection for DDoS attacks that uses neural networks best known for ”RBF”, the model classify two main classes which are Attack and Normal, if the model detects an attack the IP is immediately sent to the filtering alarm for actions to be taken otherwise the traffic goes to its normal destiny

[14] developed a method to detect DDoS attacks by the use of a combination of four methods which are ANN, decision tree, and Bayesian methods.

[15] proposes a very unique methodology which is a probabilistic neural network to classify DDoS attacks from normal traffic, This method is a mixture between Bayes rule and Radial Basis Function [15] RBF

Decision Tree Approach:

The proposed methodology in [15] consists of Storage, preprocessing module, and Detecting module as shown in Fig .2, The storage modules consists of two parts, Signature base and Learning base which contains the data set in Fig.9 used in the experiment, preprocessing module takes the data-set that process the incoming packets and removes redundant information, and then C4.5 algorithm with SNORT (signature-based detection) is used which determines whether the user is an attacker or not.

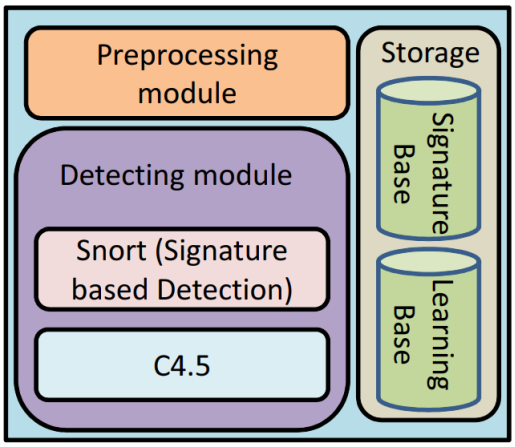

Fig. 9. Proposed Model

Simulation Results: In this paper [16] the author compared his results with Naive Bayes classifier [17] for detecting the anomalies, The author instead used a signature-based detection approach

There are no public cloud attacks in [16] done because of many reasons, the first one that will make an effect on their infrastructure which is not ethical, The second and the most important reason that these cloud providers has a high level of security layers so it is nearly impossible to attack their systems to get some results.

The whole paper relies on a fully simulated attack by the use of many components.

Land	Service	Protocol	Flag	TTL	Class
0	http	TCP	SYN	TTL≥128	Normal
0	http	TCP	SYN	TTL≤128	Attack
1	ftp	UDP		TTL≥128	Attack
0	smtp	ICMP		TTL≥128	Normal
0	http	UDP		TTL≥128	Normal
1	http	TCP	SYN	TTL≥128	Attack
0	http	UDP		TTL≥128	Normal
1	ftp	ICMP		TTL≥128	Normal
1	ftp	ICMP		TTL≥128	Attack
0	ftp	ICMP		TTL≤128	Attack

Fig. 10. Training Data set Sample

Oracle Virtual Box is used to run the simulation as the foundation environment for all the systems. In addition, It is necessary to use open-source software to build the cloud inf infrastructure on it using OpenStack Juno. The generation of the attacks was addressed by the use of Wireshark and other tools.

Method used	Correct Classification (%)	Detection Time (S)
Naive Bayesian	91.4	1.25
C4.5	98.8	0.58
K-Means	95.9	1.12

Fig. 11. Results in various Durations

According to The results in Fig.4 C4.5 is the best attack detection model in terms of running time as it is the fastest model with 0.58 seconds, In addition, C4.5 is the best in terms of correct classification as it has 98.8 accuracy whether the nearest model was K-Means with 95.9 accuracy with 3% away from C4.5.

From all of these results, C4.5 is the best method used.

	TP	FP	TN	FN	F-Measure
Naive Bayesian	282	28	253	25	0.914
C4.5	298	2	270	5	0.988
K-Means	285	24	264	0	0.959

Fig. 12. F1-Score Comparison between Classifiers

In Fig. 5 it shows other metrics in consideration like F-Measure, C4.5 have the highest F-Measure rate with 98.8 which means that it could differentiate between both classes (Attacker and Not Attacker) very well.

B. State of The Art

The three papers [10], [8], [16] have introduced many approaches to address the Detection and Classification of DDoS Attacks, some approaches as [7] relied on some rule-based like signature-based, some approaches used classical

machine learning models, while others used Deep learning models, some approaches used statistical models as well.

We cannot say that a specific paper was the best paper to address the problem, Although we can make use of some advantages of each approach, For example in [16] The author has used signature-based approach using decision tree which we think does not generalize enough however it has very quick detection time, Moreover in [7], They figured out that SVM has the best accuracy with 99% accuracy with high F1-Score however they didn't show whether their approach has drawbacks or not, hence we will try their approach on our data-set and see whether it gives us satisfying results or not.

Our initial methodology that we will use a decision tree in the first defense layer due to its high speed which was proven in [16] if the model, in the second layer we will use a combination of SVM, Random Forest, and Naive Bayes and apply max vote on the results from all used models to always get the best results and we will keep improving to enhance this methodology as we go.

III. Methodology

Our methodology pipeline involves 4 major stages: data collection, data preparation, modeling, and evaluation. The whole pipeline steps are shown in Fig. 13.

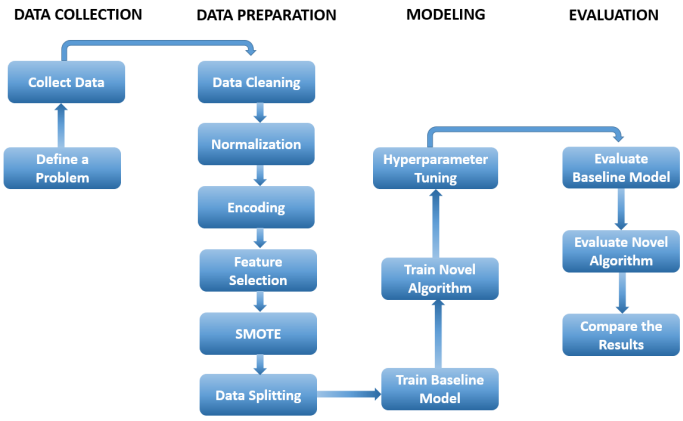

Fig. 13. Methodology pipeline

A. Data Collection

In this project, we use a dataset provided by the University of Newbrunswick called CICDDoS2019. [18] In the creation of this dataset, the creators first reviewed existing datasets, then proposed a new taxonomy regarding the DDoS attacks because of the continuous development in the attacks' types and methodologies there's a continuous need to come up with new taxonomies. Afterward, they generated the new dataset and called it CICDDoS2019, which they claim remedies all current flaws.

The data was created after analyzing novel DDoS attacks carried out using UDP or TCP protocols at the application layer. They initially split the attacks into two categories: Reflection-based and Exploitation-based attacks as shown in Fig. 14. The two categories of attacks can be launched through the application layer protocols. However, the subcategories they contain is the key difference. As reflection-based attacks include DDoS types of MSSQL, DNS, LDAP, and others. While exploitation-based attack includes subcategories of SYN flood, UDP flood, and others.

The original dataset which includes the results of the network

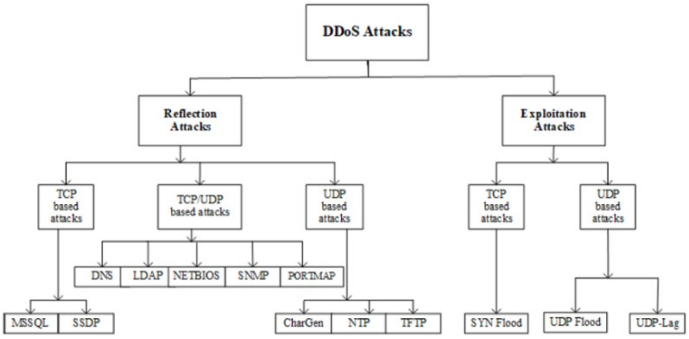

Fig. 14. DDoS Attack Categories

traffic analysis using CICFlowMeter-V3 (A network traffic flow generator and analyzer) contained 80 features extracted by the analysis tool. They contain information about the packet, the flow, network features, and much more. The labels consisted of 11 categories. All stored in CSV files of over 20GB memory space. Each file contains a binary dataset of the subcategory versus the benign flow. Our project's code is developed on google colab so in order to serve our purpose while utilizing the available memory and compute resources we decided to use a combined subset of the original data. The classes are limited to 6 subcategories which are a mix of the two main categories: Benign class, DrDos_DNS, DrDos_LDAP, DrDos_MSSQL, DrDos_NTP, and DrDos_UDP. 10,000 rows are collected of each one of 5 types of attacks in our project. Summing up to around 50,000 records.

B. Data Preparation

1) Data cleaning: The raw dataset has many unsuitable-to-train columns. For example, some feature names include spaces, some columns have a single value throughout the entire set (Protocol,..), some have nulls, infinity values. So data cleaning is applied by removing all the corrupted columns and editing the corrupted rows when possible. Also, columns like ID columns (FlowID,..) are dropped because they have no significant impact on the labels.

2) Normalization: The features of the dataset have scales very different from each other, so in order to reach more robust results and make sure that all the features are measured on the same scale so they are equally important, z-score normalization is applied to the data.

3) Label encoding: Label encoding is performed on categorical features as well as on the target label to convert them to numeric form to be able to feed them into the models.

4) Feature selection: The best 40 features are selected out of the 80 existing features based on the mutual information scores between variables. An other number of features are tried like 20 and 30 features, but it turned out that 40 features get the best performance.

5) SMOTE: Data is examined and it is found out that it is quite imbalanced as shown in Fig. 15. So SMOTE oversampler is applied to improve the results. Fig. 16 shows the data distribution after SMOTE.

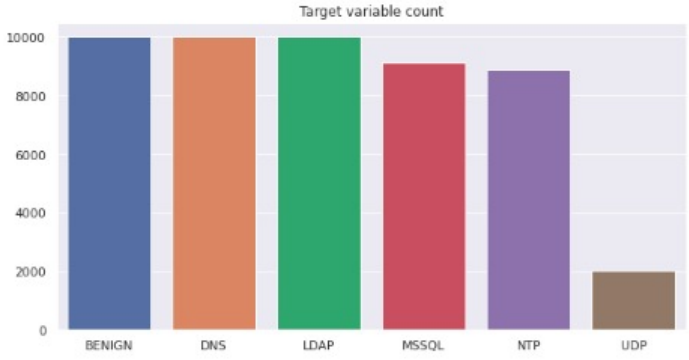

Fig. 15. Imbalanced Data Distribution (Before SMOTE)

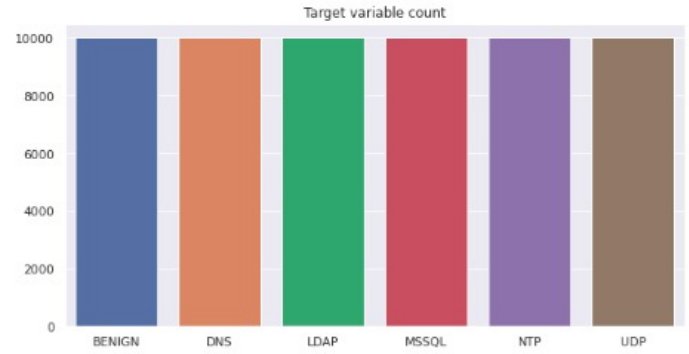

Fig. 16. Balanced Data Distribution (After SMOTE)

6) Data splitting: Data is split into training and testing sets with a ratio of 80:20. At this point, the data becomes ready to get into the next phase.

C. Modeling

Our main methodology is to use the strong aspect of each model to get the best solution possible. In our approach, we try to divide our problem into several stages to be able to conquer the problem suitably. Our approach consists of 3 stages, each stage tackles a certain problem. the first stage works on eliminating the possibility of normal data. The second stage works on differentiating between different types of attacks. The third stage works on a further specific classification of a couple of classes that do not get clear differentiation between them in the previous stage.

1) Stage one: (first defence line): The first stage aims at getting the fastest response possible. The idea behind our concern is that we need the normal data to be passed quickly without any delay or further analysis. Also, normal data samples cause a high imbalance as the ratio between normal data samples and each type of attack is very high. That's why we use binary classification in our first stage between benign and attack data. We use SVM model for this stage for its high performance in binary classification. If the input data is classified as benign no further detection is applied. Therefore, only data detected as attacks are passed to the next stage.

2) Stage two: (second defence line): The second stage targets classifying each type of attack. Identifying the type of attack is important in order to be able to eliminate the problem correctly. That's why the second stage has attack data only as an input to it with the elimination of normal data from the training data to maintain a high focus on attacks types only. Our model choice for this stage is a Random forest classifier as it uses an ensemble technique to conquer our problem which is known for its good performance in multinomial classification problems. The classification of the attack type allows the system to follow suitable procedures to prevent the occurrence of the attack and future protection from this type of attack due to the gathered data from it.

3) Stage three: (third defence line): The third and final stage works on enhancing our approach's performance. It targets certain two types of attacks from five that didn't get identified properly in the previous stage. Therefore, Only data classified as one of these two types will enter this stage for further analysis.The model used for this stage is the decision tree as it is known for its high clarity and deep analysis of the problem. Also, it is known for its good performance in binary classification.

Hyper-parameters Tuning:

hyper-parameters tuning is a very important stage in which the model is being refined to give the best results, Each model has many hyper-parameters to be changed so we will discuss each model and its hyper-parameters used.

Hyperparameter Type	Value
C	1.0
kernel	rbf
degree	3
gamma	scale
coef0	0.0
shrinking	True
probability	False
tol	1e-3
cache size	200
class weight	None
max iter	-1

TABLE I
SVM HYPER-PARAMETER USED

a) Stage one: here SVM is used as the first line of defense to classify between attacks and non-attacks.

Because SVM with the default hyper-parameters gives the best results with 100% accuracy, no hyper-parameter tuning

is done and the default hyper-parameters used are shown in table 1.

b) Stage two: Random Forest is implemented as the second line of defense to classify between the 5 types of attacks.

Random Forest has many hyper-parameters but the most effective two hyper-parameters here are estimator number and max-depth.

The number of estimators is the number of trees that the model has. Random forest is a bagging algorithm that uses multiple models to predict upon all these models, Here as shown in Fig. 17. the model is tested with a set of an incremental number of estimators starting from 50 estimators and ending with 400 increasing with 50 estimators on each step. It was clearly shown that when the number of estimators is 250 the model has the best f1-score with 0.9265.

Fig. 17. Number of estimators tuning on Random Forest

On the other hand, as the Max depth is the maximum depth of the tree, if it increases too much the model will be overfitted on the data and if it is small, the model would be very shallow and will suffer from underfitting problems. So a number in between that could give us the best results. The model is compared with different values of max depth starting from 1 to 14. It is clearly shown in Fig. 16 that the model gives the best f1-score results when the max depth is between 11 and 14 but the max depth with 14 is the best.

c) Stage three: In this stage, decision tree is used as the final model to classify between the two types of attacks that the model cannot distinguish between very well.

In this model two types of hyperparameters are tuned, the first one is Max Depth as it indicates the depth of the tree, a range of max depth between 1 and 14 are tested using f1-score. As stated in Fig. 19. the model in max depth of 8 and 9 gives the best results with a 0.87 f1-score.

The second hyperparameter is max-features which is the number of features that the model considers while searching for the best split. A wide range of max-features numbers is used between 1 and 40 features, the results are almost steady after max-features of 10, so 15 features are used which give the best results with 0.87 f1-score.

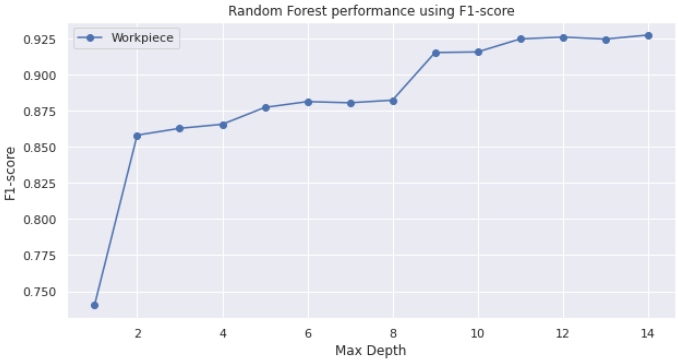

Fig. 18. Max Depth tuning on Random Forest

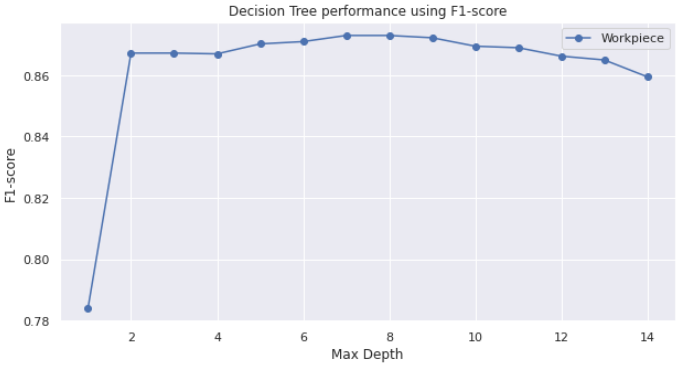

Fig. 19. Max Depth tuning on Decision Tree

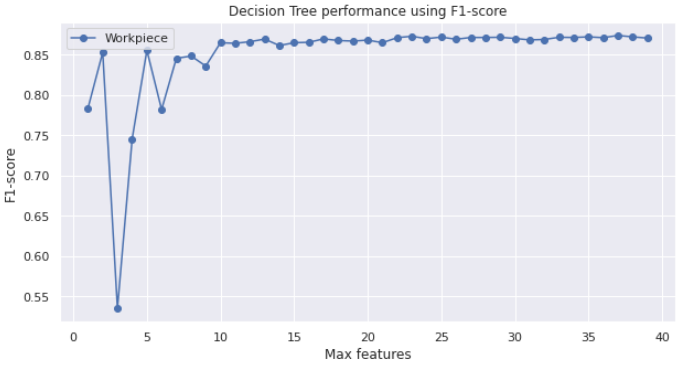

Fig. 20. Max features tuning on Decision Tree

D. Experimental Setup and Evaluation

Both models - the baseline and the novel models - are trained on the same training dataset and also tested on the same testing dataset. The same metrics are also used to evaluate the performance of both fairly.

1) Baseline Model: Random forest algorithm is selected to be the baseline for our experiment. According to the related work and state of the art, it produces very good results in

DDoS attack detection. Moreover, Random Forest, in general, is widely used due to its flexibility and ease of use.

2) Testing Methodology: to evaluate the two models hold-out method is used, where they are trained on 80% of the data and tested on the remaining 20% unseen data . The metrics used are F1-score, precision, and recall. Accuracy may not be a good metric for imbalanced datasets, but since the dataset is oversampled using SMOTE, it becomes now balanced so accuracy can be observed beside the metrics stated above.

IV. Results

A. Baseline Model

The baseline model which is implemented with Random Forest gives us 0.93 accuracy and f1-score as well.

B. Novel Algorithm

Our Methodology which is implemented on three different stages as mentioned before gives us 0.94 accuracy and f1-score as well. Although it has a slight improvement over the baseline, this improvement is over the two wrongly misclassified classes from the baseline.

Model	Accuracy	F1-Score
Our Methodology	0.94	0.94
Random Forest	0.93	0.93

TABLE II
RESULTS COMPARISON BETWEEN OUR METHODOLOGY AND THE BASELINE

C. Discussion

The results show better performance for the combination scheme over the baseline Random Forest model. These results are considered logical and resemble what was expected while designing our methodology. This is because the problem is divided into 3 stages, and at each stage, a suitable model is implemented for that specific task. In the first line of defense stage, SVM is used as it is considered one of the best algorithms in binary classification tasks. It basically tries to find the best hyperplane separating the data points of the two classes. In the second stage, detecting the type of attack, random forest is used which performs great on multi-class classification tasks. In the third stage, a good binary classification algorithm is also needed to separate between the 2 classes whose distributions are quite similar, so a decision tree is used at that stage. Therefore, this combination outperforms the baseline which is basically a random forest to detect all the 6 labels at once.

Advantages:

- The new method involves three models, where each on of them handles a specific task so it gets higher overall accuracy and F1-score.
- It first classifies whether it is attack or not 100% accurately and fast. So if it is classified as benign, it does not need to go further in the next steps.

Limitations:

- It could be more difficult to train 3 different models at each step than training only 1 model for the whole task.
- The models need to be retrained whenever a new class of attacks appears.

What we can learn from this experiment is that combining different models in a suitable way may produce better results than using a single model.

V. Conclusion and Future Work

DDoS attacks are critical threats that can cost organizations huge amounts of money and disrupt the availability of networks and devices. Since detection of this type of attack becomes challenging, a novel methodology is implemented to tackle the problem. An ensemble of three different algorithms is combined together such that each one handles a different stage of the problem on which it performs best. The results show that this methodology gets better results than the baseline model (random forest) where it scores 94% accuracy and F1-score, while the baseline scores 93% accuracy and F1-score. This shows that combining schemes may produce better results than single models.

For future work, we can try different approaches that emphasize the difference between several DDOS attacks. In model selection, other models can be applied especially deep learning models like LSTM, where it can recognize the patterns of different types of attacks. For data pre-processing, new features can be extracted using statistical techniques to give us better insights into the similarities and differences between different types of DDOS attacks. Also, Adaptive learning strategies may be used especially contextual approaches where each model detects a certain type of attack and each model is selected according to the input data features. By that, better accuracy can be achieved and new types of data can be also identified.

References

[1] S.Y. Nam, T. Lee, Memory-efficient IP filtering for countering DDoS attacks, in: Proceedings of the 12th Asia-Pacific Network Operations and Management Conference on Management Enabling the Future Internet for Changing Business and New Computing Services, APNOMS'09, Springer-Verlag, Berlin, Heidelberg, 2009, pp. 301–310.

[2] Maciá-Fernández, Gabriel, Rafael A. Rodríguez-Gómez, and Jesús E. Díaz-Verdejo. "Defense techniques for low-rate DoS attacks against application servers." Computer Networks54.15 (2010): 2711-2727.

[3] Vijayalakshmi, M., S. Mercy Shalinie, and A. Arun Pragash. "IP traceback system for network and application layer attacks." Recent Trends In Information Technology (ICRTIT), 2012 International Conference on. IEEE, 2012.

[4] Dantas, Yuri Gil, Vivek Nigam, and Iguatemi E. Fonseca. "A selective defense for application layer ddos attacks." Intelligence and Security Informatics Conference (JISIC), 2014 IEEE Joint. IEEE, 2014.

[5] Alkasassbeh, Mouhammd, et al. "Detecting distributed denial of service attacks using data mining techniques." International Journal of Advanced Computer Science and Applications 7.1 (2016).

[6] Livadas, Carl, et al. "Usilng machine learning technliques to identify botnet traffic." Local Computer Networks, Proceedings 2006 31st IEEE Conference on. IEEE, 2006.

[7] Wani, A. R., Rana, Q. P., Saxena, U., & Pandey, N. (2019). Analysis and detection of DDoS attacks on cloud computing environment using machine learning techniques. Paper presented at the Proceedings - 2019 Amity International Conference on Artificial Intelligence, AICAI 2019, 870-875.

[8] Amjad, Aroosh, et al. “Detection and Mitigation of DDoS Attack in Cloud Computing Using Machine Learning Algorithm.” EAI Endorsed Transactions on Scalable Information Systems, vol. 6, no. 23, European Alliance for Innovation (EAI), 2018, p. 159834–, doi:10.4108/eai.29-7-2019.159834.

[9] Corrêa, João Henrique, et al. “ML-Based DDoS Detection and Identification Using Native Cloud Telemetry Macroscopic Monitoring.” Journal of Network and Systems Management, vol. 29, no. 2, Springer, 2021, doi:10.1007/s10922-020-09578-1.

[10] Francisco Sales de Lima Filho, Frederico A. F. Silveira, Agostinho de Medeiros Brito Junior, Genoveva Vargas-Solar, Luiz F. Silveira, ”Smart Detection: An Online Approach for DoS/DDoS Attack Detection Using Machine Learning”, Security and Communication Networks, vol. 2019, Article ID 1574749, 15 pages, 2019.

[11] Q. Liao, H. Li, S. Kang, and C. Liu, “Application layer ddos attack detection using cluster with label based on sparse vector decomposition and rhythm matching,” Security and Communication Networks, vol. 8, no. 17, pp. 3111–3120, 2015.

[12] R. Karimazad and A. Faraahi, “An anomaly-based method for ddos attacks detection using rbf neural networks,” in Proceedings of the International Conference on Network and Electronics Engineering, 2011, pp. 16–18.

[13] J. Li, Y. Liu, and L. Gu, “Ddos attack detection based on neural network,” in 2nd International Symposium on Aware Computing (ISAC),. IEEE, 2010, pp. 196–199

[14] V. Akilandeswari and S. M. Shalinie, “Probabilistic neural network based attack traffic classification,” in Fourth International Conference on Advanced Computing (ICoAC). IEEE, 2012, pp. 1–8.

[15] J.-H. Chen, M. Zhong, F.-J. Chen, and A.-D. Zhang, “Ddos defense system with turing test and neural network,” in IEEE International Conference on Granular Computing (GrC). IEEE, 2012, pp. 38–43

[16] Lonea, Alina & Popescu, Daniela & Tianfield, Hua. (2013). Detecting DDoS Attacks in Cloud Computing Environment. International Journal of Computers, Communications & Control (IJCCC). 8. 70-78. 10.15837/ijccc.2013.1.170.

[17] W. I. D. Mining, “Data mining: Concepts and techniques,” Morgan Kaufinann, 2006.

[18] University of Newbrunswick DDoS2019 dataset: https://www.unb.ca/cic/datasets/ddos-2019.html