An Evaluation Framework for Machine Learning Methods in Detection of DoS and DDoS Intrusion

Temechu Girma Zewdie
Computer Science and Engineering
University of the District of Columbia
Washington DC, USA
temechu.zewdie@udc.edu

Anteneh Girma (Ph.D.)
Computer Science and Engineering
University of the District of Columbia
Washington DC, USA
anteneh.girma@udc.edu

***Abstract* —** A distributed denial-of-service (DDoS) and DoS attack are the most devastating and expensive attacks among various cyber and network attacks [1] [2] . Coupled with the fact that launching such attacks could be relatively easy, it makes it a big problem in the realm of Security and Cyber Space in general. However, with the advent of advanced Artificial Intelligence / Machine Learning(AI/ML) methods and tools, we explore different research techniques and methodologies to find a better detection accuracy result and prevent many different kinds of Attacks and Intrusions. During the research process, we will address Analytical and Computational challenges, Feature Selection issues, and Machine Learning Models while paying particular attention to Feature Engineering by using Mutual Information and Principal Component Analysis in the feature construction process. Moreover, K-Nearest Neighbors, Decision Trees, Random Forests, and XGBoost for Classification are used. In General, this study will target to analyze the ability of these methods to detect DoS and DDoS attacks while also examining the capacity of the ways to distinguish between different kinds of these attacks. Finally, the research investigates and proposes a framework for simultaneous evaluation of different Machine Learning methods in detecting DoS and DDoS.

Keyword—Cyber Security, Cyber Space, Decision Tree, DDoS, Dos, Feature Selection, Machine Learning, Principal Component Analysis, KNN, Random Forest, Mutual Information, XGBoost

I. INTRODUCTION

Dos Attacks are primarily a category of Tactics used to disrupt the traffic to a specific server. The malicious agent will cause the server to become temporarily unavailable or unresponsive to other users. It can be imagined as something clogging up the traffic in a road causing jams. Victims of such attacks can include computers, servers, or other networked resources such as IoT devices. DDoS attacks are distributed forms of DoS attacks. These tactics are usually used in tandem. DoS attacks can be precursors to DDoS attacks, and the latter is often followed by the latter. These sorts of attacks are carried out through a Network of Machines. DoS and DDoS attacks are relatively simple but extremely powerful and presently remain an immense threat to network security, and organizations spend much effort trying to address the issue. DDoS exploited the inherent nature of the Internet, and it's an open-source model, which is also its most significant advantage.

Nowadays, software and tools are developed and distributed to manage numerous types of attacks. These allow individuals to sort episodes quickly while providing a user-friendly experience. However, the attacks make the problem even worse.

This paper will investigate the use of Machine Learning methods and techniques in identifying and defending against these attacks. We will start with a definition and a brief taxonomy of these attacks. We will then outline the steps and procedures needed to use Machine Learning methods. Furthermore, we will describe some challenges and provide experimental results that validate our processes and procedures.

The paper will identify these attacks with multiple methods for the Feature Engineering phase. First, we put a set of essential features in Machine Learning models. Then, we will use methods of Dimensionality Reduction and test our main Machine Learning models to use these features.

Finally, we will use K-Nearest-Neighbors, Decision Trees, Random Forests, and XGBoost and provide experimental results. In this research, Finding the suitable dataset itself has been identified as a challenge in the field. We will be carrying out these experiments using the CIC-IDS-2017 dataset.

II. DOS AND DDOS ATTACKS

A. Definition

A DoS attack is an attack that can render a network incapable of providing regular services [2]. The purpose of a DoS attack is to obtain access to a network resource that has been intentionally blocked or weakened through the actions of a malicious agent. Attacks don't necessarily damage network systems or data permanently, but they temporarily compromise the availability of the resources [1]. DDoS attacks occur when multiple systems have coordinated to attack concurrently to aggravate the offense. These attacks hit the target system simultaneously. A malicious agent could obtain other systems to carry out the intended attack.

B. Taxonomy of the attacks

The most common DoS attacks target the computer network bandwidth or connectivity [1]. These attacks compromised bandwidth "flood" the network with many network requests. So, the system will not provide resources to innocent users. It can result in degraded performance or even complete shut-down. Connectivity attacks flood a network with a high volume of connection requests, and the computer can no longer respond to innocent requests. The attacks can also be categorized based on the protocol (based on the 7-layer OSI scheme) they attack. We can also further break up each kind into subcategories [3].

Application Layer:

- Some attacks render a network of machines out of order by taking advantage of specific bugs or weaknesses in the network applications hosted by the target or by requesting unnecessary resources. For example, the attacker may have launched computationally expensive requests and unnecessarily taken up resources.

Protocol Level Attacks:

- At the Operating system (OS) level, DoS attacks the target and takes advantage of the weaknesses in protocol implementation. The attacker may send invalid or unnecessary protocol level requests such as ICMP (Internet Control Message Protocol) or IGMP (Internet Group Management Protocol) requests clogging up the system and halting regular protocol activity.
- Protocol Feature Attacks: these attacks exploit the inherent structure of specific standard protocols and their features. Some exploit the system by taking advantage of certain features of the IP address by spoofing their internet protocol. Others can target the structure of the DNS by trapping the victim into caching false records.

Volumetric Attacks:

- Network Device Level attack is an attack caused by exploiting the weaknesses in firmware or by trying to exhaust the hardware resources of the network. I.e., One may drain the routers, hard drives, or other network hardware resources.
- Data Floods attack may include sending substantial data segments through the network bandwidth, thereby clogging up the network with the unnecessary data flow.

The attacks do not necessarily disrupt network services completely, and they may only degrade them. We can classify these attacks based on various issues such as degree of automation or rate dynamics, yet the above is adequate for our purposes.

III. MACHINE LEARNING IN DETECTION OF DOS AND DDOS

To defend a network of systems against any intrusion or potential Cyber Attack, including DoS, we must first detect the attack and then act in time to regulate the situation. Machine Learning is used for cybersecurity measures, particularly in three essential domains: anomaly detection, intrusion detection, and misuse detection [3]. The main problem of Machine Learning in detecting DoS is that of Classification, which is a Supervised Learning problem.

In a Classification problem, we have data of previously identified attacks that can serve as blueprints for identifying new ones.

A. Review of relevant literature

The issue of intrusions has been around since the advent of computation. However, with the advancement in communication technology, this problem has been severely exacerbated. The term intrusion refers to "any unauthorized access that attempts to compromise confidentiality, integrity, and availability of information resources" [4].

Research in the field of detection of Cyber Intrusion has taken various directions. Many have applied statistical models such as Logistic Regression. Others have employed Neural Network models, yet others have used other Machine Learning Classifiers such as Bayesian classifiers, Support Vector Machines, and Lazy Learners.

In [5], Yan et al. propose a system that detects botnets in real-time by only using features from higher-level layers of the OSI. T. Cai and F. Zou [6] presented some features of HTTP Botnet and designed a detection method using clustering based on them. Chien-Hau Hung, Hung-Min Sun created a Botnet Detection System Based on Machine-Learning Using Flow-Based Features. In [7], F. V. Alejandre attempts to detect botnets using Machine Learning and the Evolution of Genetic Algorithms to select Features. In [3], Sofi, Mahajan, and Mansotra investigated the use of Machine Learning Techniques for the Detection and Analysis of Modern Types of DDoS Attacks. In [8], the authors proposed a Random Forest model for detection. The authors of [9] utilize the flow-based features of existing botnets and select 21 features for machine learning, and the outcome of the average detection rate was about 75%.

B. Research Methodology

In a Machine Learning project, feature engineering is a crucial step requiring domain knowledge to extract features from raw data. Once pre-processing is accomplished, Feature Engineering is the next essential step in Machine Learning methods. We often don't have the computational ability to use all features of datasets in our models, and we must first transform or subset a set of essential attributes or features to use them in our models. Many different Supervised Learning methods can use as a model. The selection and comparison of these methods and algorithms are essential in finding the most efficient computationally and accurately system.

Apart from previous work, in this paper, we will be using different kinds of Feature Engineering (Feature Selection) approaches and testing them against other Machine Learning (Classification) algorithms to compare the results. We will be systematically looming the matter so that our work can be used as a framework for evaluating and comparing different methods in Pre-Processing, Feature Engineering and Modeling, and Testing methods to detect DoS and DDoS Intrusion. Our Framework can be adapted and extended in other Cyber Intrusion solutions that consistently achieve the best results for Feature Selection and investigate whether we can find a Classification method that consistently achieves the highest regardless of the Feature Selection method. If the result is that, we can further study the issue by focusing on the most accurate solutions. Finally, our primary purpose is to propose a framework for processing data and evaluating models and Feature Engineering methods.

C. The Datasets

Finding a suitable dataset for our purposes itself is a challenge. We need to find datasets with all the features of the communication packets, and also each instance must be

correctly identified beforehand as an attack or benign activity. Most of the datasets online lack traffic diversity and volume. Besides, they do not cover the variety of known attacks, while others anonymize packet payload data, which cannot reflect the trends. Moreover, some are lacking feature sets and metadata [10] .

For this research paper, we will be using the CICIDS2017 dataset from the University of New Brunswick. This dataset is an Intrusion Detection and Prevention dataset, and precisely We will be using the 'Wednesday-WorkingHours' data. The dataset has 79 columns of data. One of which is the label that indicates if the instance is an 'attack' or 'Benign' communication. It also indicates what kind of attack tools was used. The attacks are carried out using the following popular tools:

- Hulk Dos
- Slowloris Dos
- Slow Httptest
- Dos Goldeneye

IV. Pre-Processing

In this pre-processing stage, a data clearance must be done before starting our research. I.e., we have to handle those data with unacceptable values for our algorithms to use. I.e., Nan (Not a Number) values or Inf (Infinity) Values have to rectify before we start.

There are many methods to handle such issues, such as dropping values, imputation, and extended imputation. For simplicity and better accuracy, we will drop instances with unacceptable values. An attribute containing instances with 'Inf' values such as 'Flow Bytes/s' and 'Flow Packets/s' will drop before continuing to the next step.

V. Feature Engineering

Feature Engineering refers to creating or selecting features from the data to be used in Machine Learning. In this work, we will be using three methods for Feature Engineering. After removing features with zero Variance, we will be using Mutual Information and Principal Component Analysis (PCA) to reduce the features of the datasets for use in the Machine Learning models. We will either transform these features using the PCA or select the most significant features using the other two methods.

A. Removal of Features with zero Variance

Before we proceed, we will remove the columns from the dataset with zero Variance. Columns with constant values don't change from instance to instance. Variance thresholding itself is a technique that can use in feature selection. A threshold is often chosen, which will be the cutoff point to include any features. If a particular column has a higher variance than the cutoff, it should be included in the model; otherwise, ignored. Here we will not be using Variance to select features for our models. Our cutoff variance will be zero, meaning that we will only drop columns of data with Variance zero, that is, a column with constant values.

B. Mutual Information

As referred to, an Information Gain measures how a target variable, which is the kind of DoS Attack, is dependent on other variables in the data. We will be using a sklearn-learn library to compute the Mutual Information, and we will use that to identify the most minor and most essential features for our purpose.

Sampling

Computation of Mutual Information can be quite expensive. It can take quite a long time to compute the Information of each Feature in our DDoS dataset. In order to achieve this, we will be using sampling. We will sample the dataset randomly at a fraction of 0.05 and then compute the Mutual Information for Features of the sample dataset. The values computed for the sample shouldn't be much different from the actual population as the sampling was random.

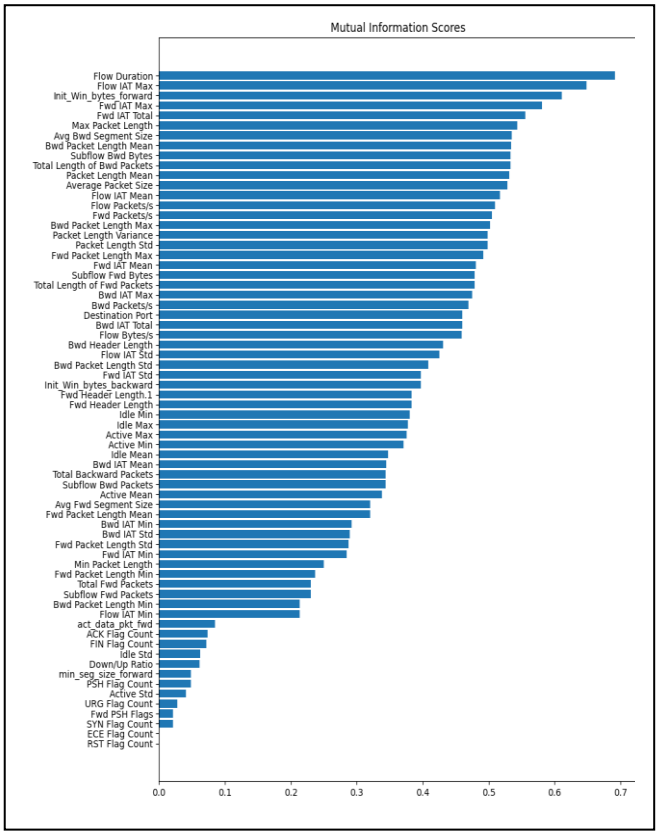

Fig. 1. Mutual Information

We will be using a range of numbers for the number of features selected for our purposes. We will be treating the number of the columns used as a hyper-parameter that will help tune in subsequent sections.

Here we will define some of the topmost and necessary features in the data. The definition will give us insight into what characteristics a potential intrusion may have also will provide us with a sense of what will be essential to us in our modeling [11]:

_*Flow Duration:* The duration of communication flow. Which means the temporal length of the flow. It means the total time of the connection from beginning to end. It has been used a lot in Intrusion Detection.

_*Fwd IAT Max:* The Inter-Arrival Time is the amount of time that elapses after receiving a packet until the next one arrives. Fwd IAT Max refers to the maximum of all IAT's forward direction.

_*Packet Length Mean :* Average *length* of packets transferred in the *connection* flow.

_*Subflow Bwd Bytes:* The average number of bytes in a sub-flow in the backward direction.

_*Flow IAT Mean:* Mean of Inter-Arrival Times in the flow.

_Packet Length Std: Standard deviation of the length of packets in the connection.

_*Fwd Packet Length Max:* The maximum length of all packets in the forward flow.

_*Fwd IAT Mean:* Average mean of Inter-Arrival Times in the forward direction.

To interpret the meaning of this, we will first have to understand the importance of features mentioned above to our Intrusion Detection models. These features suggest that the most decisive features that can determine the banality of malice of a connection concerning to Dos attacks are usually related to durations of flow, duration of Inter-Arrival Times, and length of packets. We can further statistically analyze these features to instigate further insight.

C. *Principal Component Analysis*

PCA is a method for transforming the features of a dataset into a new set of features. It is a mathematical mapping that will map a location of points from one space to another. However, the features will be ordered decreasingly regarding importance (i.e., the first features will carry more weight than those coming after). Thus, reducing the need for too many features. We can capture the essential features of the data using fewer features in a different space. Figure 2 shows the variance explained by the first ten components after transformation. We can see from this and figure 3 that we capture nearly all of the variance in our data using only a few (perhaps 5 or 6) components.

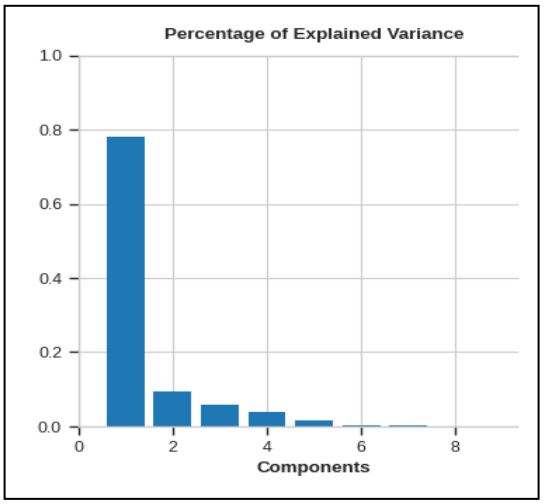

Fig. 2. Percentage of Explained Variance

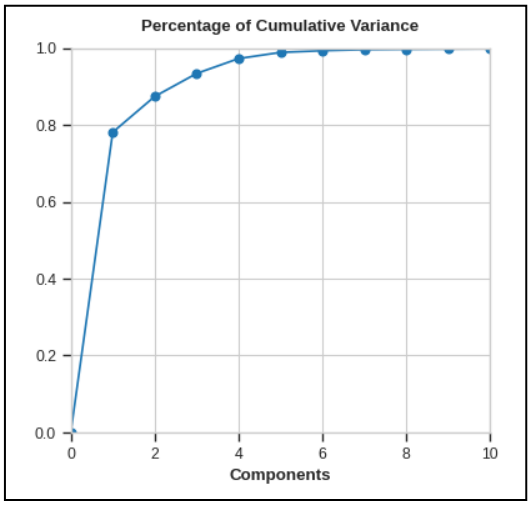

Fig. 3. Cumulative Variance

VI. MACHINE LEARNING

In this section, we will be using Machine Learning algorithms to classify the activities, i.e., primarily address a classification problem. Each instance of the data will be labeled either a benign activity or labeled as an attack.

A. *Decision Tree Classifier*

The Decision Tree Classifier is one of the most known and used Machine learning algorithms. The simplicity and efficiency make it a great candidate in any application. A decision tree is a classifier for determining an appropriate action (among a predetermined set of steps) for a given case [12]. It helps for Classification and Regression purposes. The algorithm works by creating certain simple decision rules that predict the value of a target variable. Representations of the Decision Tree can look like an inverted tree hence the name.

An Architecture of the Decision Tree

A decision tree has three kinds of Nodes. The Root Node is the highest, and It has no incoming Edge but can have none or more outgoing Edges. Internal Nodes have exactly one incoming edge and two or more outgoing Edges. Leaf Nodes have precisely one incoming Node and no outgoing Edges [13].

B. Random Forest Classifier

The Random Forest is an Ensemble of Decision Trees. Random forests [14] are the most popular bagging technique in machine learning, where we use decision trees as the base models. In other words, a random forest consists of many iterations of the Decision Tree, each of them constructed using a bootstrap sample. The outcome will usually be the average of all the outputs of the Decision Trees. The Random Forest Classifier achieves very highly and is very efficient due to its simplicity.

C. k-Nearest Neighbors Classifier

Perhaps the simplest of all Supervised Learning methods is the KNN. This algorithm is in the category of Lazy Classifiers. As opposed to Eager Classification, which attempts to capture deep structures in the data, this kind of algorithm imitates the nearest or most similar (in terms of Classification Features) examples. So the KNN will look for the k most similar measures in the dataset and assign the label per those instances. It could output the weighted average or simply the most occurring label.

D. XGBoost Classifier

Arguably the most accurate modeling technique for structured data. XGBoost is an optimized Distributed Gradient Boosting framework designed to be highly efficient, flexible, and portable [15]. It provides Machine Learning algorithms using the Gradient Boosting framework. It is capable of Automatic Feature selection, cleverly penalizes inaccurate trees, shrinks leaf nodes, makes use of randomization parameters, it can also utilize distributed and out-of-core computation.

VII. RESULT

Like any other scientific or engineering discipline, Machine Learning also requires experimentation. Our experiments are usually only computational. Nevertheless, it is an essential part of our process. In this research, we achieved very high accuracy results in our experiments. Here we will present these results and provide an analysis. Here, we will represent the results in two separate graphs, one for features selected using Mutual Information and one for features based on Principal Component Analysis. The accuracies shown are accuracies on the Test Set. The accuracy for each algorithm is juxtaposed alongside the others.

Two-step process of Classification

In any Classification problem, we have two steps to be accomplished: Optimization, also known as Training and Prediction. These processes are carried out on the datasets using different algorithms. The first one, also called the Optimizer, will run through the Train Set and capture its essential features. The Predictor then uses this structure to Predict the labels of new data.

We are concerned about the accuracy of prediction on new data. To test our Classification models, we will need two separate datasets. We must first Train the program on the Train dataset and then evaluate the accuracy of the Test dataset.

Train-Test Splitting of the Dataset

To Train and then Evaluate our model, we must first create two separate datasets, one for the Training and one for evaluation of the model. To achieve this, we will split our dataset into a Train and Test set using randomly chosen samples. The fraction of the data used for Testing is 30 percent, while the rest will use for Training.

A. Evaluation Metrics

The metric we use in this paper is classification accuracy on the Train set. We must keep in mind that our models also predict the tool used in the attack. The accuracy that we measure here is the Accuracy, Precision, and Recall of Classification. The latter two will use in a binary way. That is, we will be treating the output in a binary format. If the output is Benign, the binary value is set to False if it is an Attack, the binary value is set to True. Here are the formal definitions of the metrics we will be using in the present work:

$$Accuracy = \frac{Number\ of\ correct\ Predictions}{Total\ Number\ of\ Predictions}$$

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives}$$

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives}$$

In the figures below, we can see the Accuracy of the Classifiers plotted versus the number of Features chosen. There are two graphs, one for Features selected based on PCA and one for Features based on Mutual Information. These plots allow us to compare the performance of different Classifiers and also allow us to compare the performance of a single Classifier for different numbers of Features. The Classifications here are very accurate.

We can see that the KNN always has the upper hand for PCA-based Features, followed by the Decision Tree and Random Forest. The optimal number of Features we need to use here for PCA-based features is 5 or 6. Beyond that, we don't need anymore. We can see that as we were expecting from the PCA plots, we don't need more Features to capture the essential information of the data. We have effectively reduced the number of Features needed while preserving Accuracy.

Figure 4 shows the Accuracy of the Classifiers for Mutual Information Feature selection. Here we can see that Random Forest has the upper hand, followed closely by the Decision Tree and K-Nearest Neighbor. XGBoost has the lowest performance in both scenarios. Here again, we can reach near-perfect Accuracy by using six or more Features based on Mutual Information. In effect, we have chosen a small subset

of features instead of all of the classification features, improving efficiency computationally.

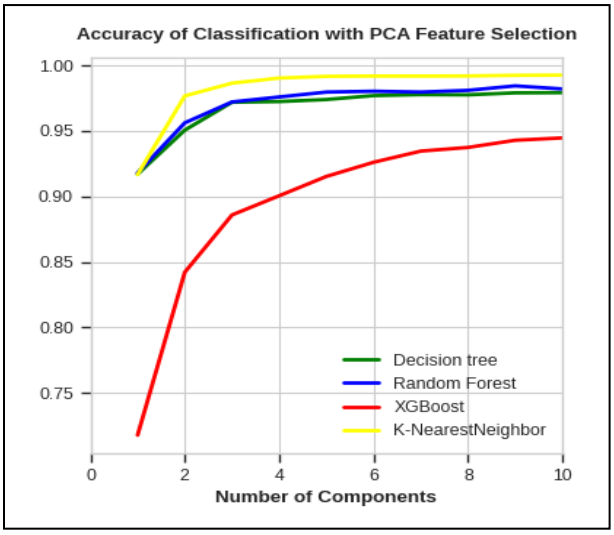

Fig. 4. Accuracy of Classification with PCA Feature Selection

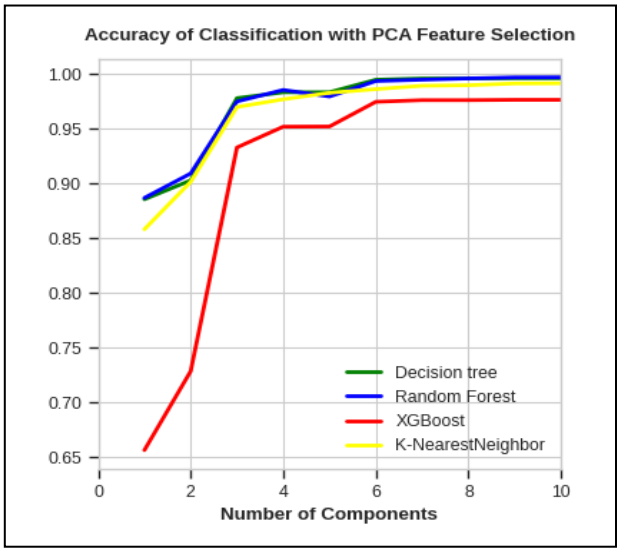

Fig. 5. Accuracy of Classification with Mutual Info Feature Selection

Here we can see the precision and recall tables of our Classifiers. Again, one table is the results for our PCA Features selection, and the other is for Mutual information Feature selection. We can generally see that precision and recall for Mutual Information based selection is always higher except for KNN. For the KNN, it seems like our precision and recall are improving with PCA Features.

Features	DTC precision	DTC recall	RFC precision	RFC recall	KNN precision	KNN recall	XGB precision	XGB recall
1	0.921936	0.95282	0.921676	0.95448	0.908277	0.969902	0.902277	0.962379
2	0.95047	0.97398	0.957333	0.97462	0.974629	0.989307	0.945196	0.977386
3	0.981242	0.97545	0.975595	0.98069	0.985818	0.993129	0.963526	0.981792
4	0.974674	0.98208	0.976446	0.98588	0.989795	0.995131	0.965017	0.986335
5	0.977144	0.98232	0.981042	0.98695	0.99131	0.995738	0.96552	0.988693
6	0.981258	0.98266	0.98191	0.98705	0.991618	0.995799	0.967784	0.989376
7	0.981142	0.984	0.979021	0.98911	0.991618	0.995768	0.968225	0.990369
8	0.980636	0.98427	0.981183	0.98895	0.991886	0.995662	0.969267	0.990635
9	0.980478	0.98684	0.987175	0.9882	0.992502	0.995753	0.97017	0.991461
10	0.981569	0.98583	0.982497	0.98925	0.99266	0.995829	0.970406	0.992894

Tab. 1. Precision and Recall of Classification for different number of Features for PCA Feature Selection

Features	DTC precision	DTC recall	RFC precision	RFC recall	KNN precision	KNN recall	XGB precision	XGB recall
1	0.920722	0.90124	0.921243	0.90266	0.846989	0.953726	0.909032	0.913216
2	0.93328	0.91596	0.935068	0.92311	0.893082	0.963077	0.928154	0.924409
3	0.980731	0.98574	0.975868	0.98593	0.970072	0.985175	0.969274	0.992758
4	0.982642	0.9947	0.982282	0.99599	0.977598	0.988496	0.973095	0.996163
5	0.982187	0.99514	0.979318	0.98962	0.983512	0.991545	0.973842	0.996049
6	0.996096	0.99645	0.993202	0.99717	0.986822	0.992629	0.975939	0.998741
7	0.996478	0.99763	0.995262	0.99716	0.990699	0.993539	0.975996	0.998696
8	0.996508	0.99762	0.996313	0.99792	0.991107	0.993948	0.975996	0.998696
9	0.996652	0.99783	0.997386	0.99823	0.992197	0.995147	0.976004	0.998749
10	0.996652	0.99782	0.997696	0.99817	0.992311	0.995276	0.976004	0.998749

Tab. 2. Precision and Recall of Classification for different number of Features for Mutual Information Feature Selection

PCA-based features and the other is for Mutual Information base features. The mistakes (or confusion) for the Classifier with Mutual Information features are lower than those with PCA-based features. In general, it seems like Mutual Information based selection is the better choice for Decision Trees in this problem.

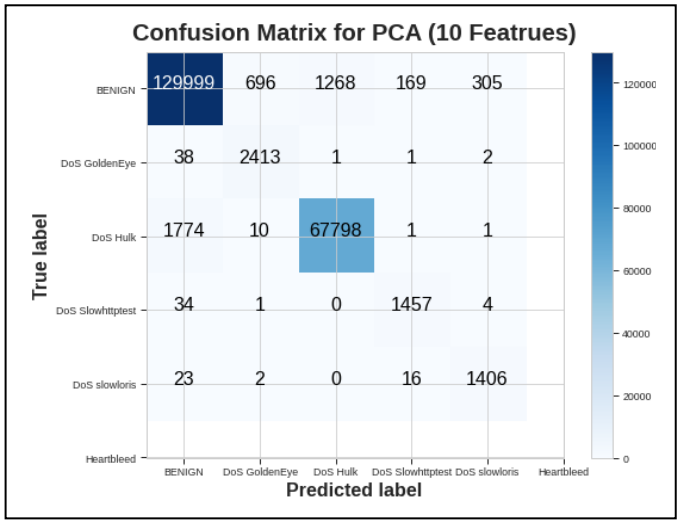

Fig. 6. Confusion Matrix of Decision Tree Classification with 10 PCA Features

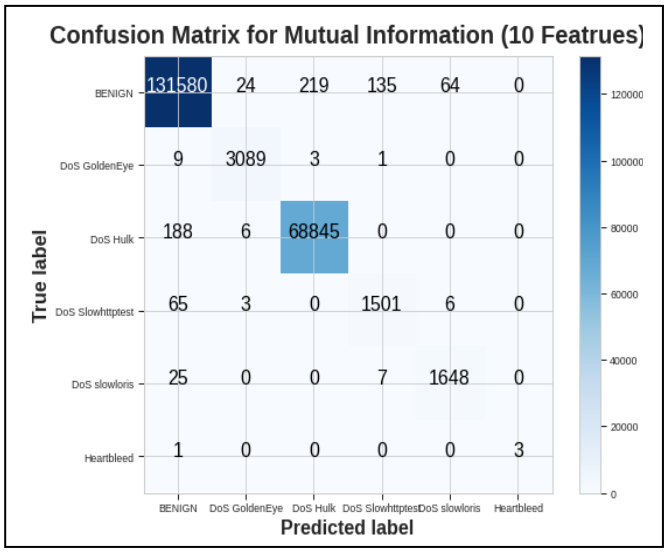

Fig. 7. Confusion Matrix of Decision Tree Classification with 10 Mutual Information Features

B. Discussion

The algorithms were evaluated, and the results were juxtaposed and plotted above gives us a chance to judge and compare the different approaches.

VIII. CONCLUSION

Herein we have shown the adequacy of our modeling and Machine Learning techniques to deal with the issue of Intrusion and, in particular, DoS and DDoS Intrusion. We have shown that our algorithms not only discern the fact that a Dos Intrusion is being made but also determine which tool was used in the attack. However, our focus was to show how we can compare different Classifiers and different Feature Engineering methods. To do that, we have demonstrated and proposed a framework for comparing these different methods. The results determined that Mutual Information is perhaps a slightly better tool, in the long run, to use for Feature Selection of such problems.

Most importantly, we achieved high accuracy results with all the algorithms, with XGBoost being the worst. Moreover, we developed and expounded a framework for analyzing, pre-processing, modeling, benchmarking, evaluating multiple Feature Engineering and Machine Learning models. In addition, we outlined all the steps needed to use Machine Learning and presented a framework upon which future research can be based. Researchers may use our framework to experimentally investigate the adequacy of their models for their specific problems. The problems don't have to be confined to DoS Intrusions, but they can be applied to similar problems.

REFERENCES

[1] C. Douligeris and A. Mitrokotsa, "DDoS attacks and defense mechanisms: classification and state-of-the-art," *Computer Networks,* vol. 44, no. 5, pp. 643 - 666, 2004.

[2] L. Stein and J. Stewart, "The Worldwide Web Security FAQ," 4 February 2002. [Online]. Available: https://www.w3.org/Security/Faq/.

[3] I. Sofi, A. Mahajan and V. Mansotra, "Machine Learning Techniques used for the Detection and Analysis of Modern Types of DDoS Attacks," *International Research Journal of Engineering and Technology (IRJET),* vol. 4, no. 6, 2017.

[4] K. Scarfone and P. Mell, "Guide to Intrusion Detection and Prevention Systems (IDPS)," February 2007. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/legacy/sp/nistspecialpublication800-94.pdf. [Accessed 4 December 2021].

[5] Q. Yan, Y. Zheng, T. Jiang and W. Lou, "PeerClean: Unveiling peer-to-peer botnets through dynamic group behavior analysis," in *Computer Communications (INFOCOM)*, 2015.

[6] T. Cai and F. Zou, "Detecting HTTP Botnet with Clustering Network Traffic," 2012.

[7] F. V. Alejandre, N. C. Cortés and E. A. Anaya, "Feature selection to detect botnets using machine learning algorithms,," in *in Electronics, Communications and Computers (CONIELE- COMP), 2017*, 2017.

[8] J. Pei, Y. Chen and W. Ji, "A DDoS Attack Detection Method Based on Machine Learning," *Journal of Physics,* vol. 032040, no. 1237, 2019.

[9] E. Samani, H. H. Jazi and A. Ghorbani, "Towards effective feature selection in machine learning-based botnet detection approaches," in *Communications and Network Security*, 2014.

[10] University of New Brunswick, "Intrusion Detection Evaluation Dataset (CIC-IDS2017)," 2017. [Online]. Available: https://www.unb.ca/cic/datasets/ids-2017.html. [Accessed 01 12 2021].

[11] H. Lashkari, "CICFlowMeter/ReadMe.txt at master · ahlashkari/CICFlowMeter.," 2021. [Online]. Available: https://github.com/ahlashkari/CICFlowMeter/blob/master/ReadMe.txt. [Accessed 12 December 2021].

[12] N. Sharma, A. Mahajan and V. Mansotra, "Identification and analysis of DoS attack Using Data Analysis tools," *International Journal of Innovative Research in Computer and Communication Engineering,* vol. 4, no. 6, pp. 11368-11375, 2016.

[13] N. Sharma, A. Mahajan and V. Mansotra, "Machine Learning Techniques used for the Detection and Analysis of Modern Types of DDoS Attacks," *International Research Journal of Engineering and Technology (IRJET),* vol. 4, no. 6, pp. 1086-1092, 2017.

[14] H. Jiang, "Machine Learning Fundamentals A Concise Introduction.," Toronto, Cambridge University, 2021, pp. 111-112.

[15] xgboost developers., "XGBoost Documentation," 2021. [Online]. Available: https://xgboost.readthedocs.io/en/stable/. [Accessed 07 12 2021].