National College of Ireland

Efficient Detection of Different DDoS Attacks using SVM, Random Forest and K-means Classifier

MSc Research Project
Cybersecurity

Tanya Stanley
Student ID: x20181744

School of Computing
National College of Ireland

Supervisor: Michael Pantridge

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Tanya Stanley
Student ID:	X20181744
Programme:	MSc in Cybersecurity **Year:** 2021-22
Module:	MSc Internship
Supervisor:	Michael Pantridge
Submission Due Date:	15/08/2022
Project Title:	Efficient Detection of Different DDoS Attacks Using SVM, Random Forest, K-means Classifier

Word Count:4182 **Page Count: 20**

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: .....

Date: 15/08/2022

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Efficient Detection of DDoS Attacks using SVM, Random Forest, K-means Classifier

Tanya Stanley
x20181744

Abstract

Distributed Denial of Service is an attack that tries to overwhelm the victim system or network with malicious traffic, endangering the service's availability. It is known to be one of the most common cyber attacks done on networks, still detecting it at an early stage has not become perfect or accurate. This study proposes three different models, namely Support Vector Machine (SVM), Random Forest Classifier and K-means to detect and classify DDoS attacks and differentiate attack traffic from benign traffic. For the purpose to classify and analyse these models, we have used three datasets each comprising of a different DDoS attack. The models are generated using Principal Component Analysis to determine the essential features and narrow down the dimension of our dataset. The models effectively classify the traffic with respect to its nature i.e., whether it is malicious or not.

1 Introduction

In the current time, protection is vital for every small thing on the Internet. The Internet offers a wide variety of information, services, and resources that enable strong connections between all the sectors. As the demand for the internet increases over time, a number of security-related problems arise. Because its utility was prioritized over its security, the architecture of the internet is mostly to blame for its vulnerabilities. Therefore, a number of assaults and threats are a cause for concern about internet security.

The most common threat to organizations is DDoS attacks. DDoS attacks target network availability by using up all of its resources, usually causes denial of service, and their frequency and volume have been rising quickly in recent years. Shorter assault durations with greater data volume are an increasingly common trend.[1]

For our research, we have used a SDN specific datasets which was produced by making use of a Mininet emulator. The main motive of this work is to implement both supervised as well as unsupervised classifiers for the detection of distributed denial of service attacks and find out which of these is the most accurate and efficient in detecting these types of attacks.

Each of our datasets is specific to one type of DDoS attack, ICMP, TCP Syn attack and UDP flood attack. All the datasets comprise of malicious and benign traffic. The benign traffic is labelled as 0 whereas bad traffic is labelled as 1. Among the 23 features in all the original datasets, some have been drawn out while the others have been computed. Switch-id, Packet count, Byte Count, Duration sec, Duration nsec, which is Duration in Nanoseconds, Source IP, Destination IP, and Total Duration are among the characteristics that were extracted. the port numbers. The number of bytes transported from the switch port is denoted by tx bytes, while the number of bytes received on the switch port is denoted by rx bytes. The date, time are displayed in the dt field after being converted to numbers, and a flow is observed every 30 seconds. Among the calculated characteristics are Packet Rate is the count of packets sent in every second. The packets that are sent in one go are Packet per flow. The no. of bytes that are transferred in a singular flow id Byte per flow.[2]

2 Related Work

An in-detail review of the previous work done on this subject is covered in this section. The background investigation of all the research done prior to ours was carried out and different approaches to the same topic are listed further.

For identifying DDoS assaults, the majority of recent work has employed datasets like the KDD Cup '99[3] dataset or the DARPA [4] dataset. As time has gone on, however, cybercrimes have been carried out in a skilful manner to enter the target area. in order to train the classifier. Using a current dataset that has a wide range of innovative attack signatures will enhance the classifier's performance.

2.1 Application of SVM for DDoS prediction

The researchers of [5] have used multiple models to get the best fit for their dataset. They created their own dataset with the help of hping3 software for simulation of DDoS attacks while normal traffic was generated by hosts present in their network. They found out that SVM was the most accurate of all.

[6] adjusts the model using the techniques he has learnt. The most effective method for identifying DDoS attacks is provided by this approach. Deep Feed Forward (DFF), as well as SVM are both used. The examination of the packets and IP addresses gives this study an advantage. However, there aren't many pcap files that show features of traffic that may be split into two timeframes. This highlights the need to evaluate an attack in an actual environment to make sure it is acceptable for the model.

In [7], Attack data is taken from the KDD99 dataset, and important attributes have been chosen based on the information gain ranking. According to experimental findings, fuzzy c-means clustering provides better categorization and is quicker than other methods.

2.2 Application of Random Forest classifier for DDoS prediction

In order to address the shortcomings of the current machine learning algorithms, [

[8] combined the model's context with its stable fit features. The three features of SIDI, SIDP, and DPDI are used to characterize the properties of TCP, UDP, and ICMP flood attacks in this work. The idea of data flow Shannon entropy is also introduced. The DDoS detection approach based on the RFC models has a greater prediction accuracy and a reduced false alarm rate when compared to HMM and SVM methods.

in [9], to identify DDoS attack, they have used a variety of machine learning algorithms. The techniques we utilize in our work—Random Forest (RF), SVM, and K-Nearest Neighbor—have shown encouraging results. For both training and testing data, had an efficiency of 99.13 percent, and 97% on all test data.

The NSL-KDD dataset is used to assess the suggested strategy in [10]. This study demonstrates that the 'Random Forest' technique that is suggested has a significant impact on the overall correctness of the analysis. For categorization, F-Measure, and MCC, this

technique has a documented accuracy of 99.9%. Comparison of several algorithms and the suggested approach reveals that the suggested technique excels in crucial assessment criteria like accuracy and F-Measure, among others. We discover that the suggested approach is effective in detecting denial of service attacks with a better degree of accuracy and fewer errors.

The researchers in [10] set up a Tomcat web server on a different computer, and the client used the DDoS attack tools Stacheldraht and Tfn to attack the server. Data from the collection was put into Bayesian network and SVM models. Because of the sets of data and characteristics, SVM outperforms the Bayesian network in terms of accuracy and processing efficiency. Because the data in this study were not sufficiently complicated to seriously test the models, it may not be trustworthy.

2.3 Application of K-means for DDoS prediction

In [11], there is a comparison between Naïve Bayes, KNN classifier and K-means models for the efficient detection of DDoS attack data is taken from the KDD99 dataset, and important attributes have been chosen based on the information gain ranking. The attack data and benign data are distinguished using WEKA tool. According to experimental findings, fuzzy c-means clustering provides better categorization and is quicker than other methods.

[12] has used K-means clustering to determine the imbalance in the traffic. A message register was used in this study to eliminate malicious packets coming in through the traffic, then evaluate the performance. Their outcome was that their model defended against different scales of DDoS attacks without hindering the service.

An innovative method for assessing the network flow proposed by the researchers in [13] is to create a matrix with the chosen traces of network. To evaluate the performances, this study uses two important aspects. They are emphasizing their efforts by using the false positive rate. Two distinct classification methods are used in this study. The two models are Nave Bayes and K-Nearest Neighbor. An enhancement in detection rate, a selection of critical features using PCA are all crucial components of this method. The restrictions are detecting the actual signatures of the attackers even in the most recent assaults.

3 Research Methodology

We made use of the SDN DDOS attack dataset in order to predict the DDOS attack. During the whole pre-processing pipeline, we carried out the separation of the original SDN dataset on the basis of different protocols in the whole dataset. There were a total of three types of protocols, TCP, UDP, and ICMP. As we split up the dataset based on the protocol, the protocol column was eliminated first from the dataset.

The cleaning of the datasets was initiated by eliminating the duplicate and erroneous entries present in it. Having cleaned the data will eventually boost output and enable you to use the best information possible when making decisions. Data processing will be unsuccessful if duplicate elements are not eliminated. To ready the dataset for further processing, this control aims to eliminate multiple occurrences of records.

The next step was standardizing the datasets. Standardizing is done in order to rectify, harmonize, and eliminate any duplication, mainly focusing on converting the data into a standard format. Data standardization can result in improved machine learning, improved data flows, and simpler law enforcement in situations when data-fed algorithms violate rights or

cause unwarranted harm. Additionally, it may promote a dispersed data gathering environment that is more competitive. [14] In order to reduce the features to produce more accurate results, we applied Principal Component Analysis (PCA) on all the datasets. The primary motive was to train the models faster as well as efficiently and also to avoid overfitting of our model. PCA is a method for lowering the complexity of certain datasets, improving interpretability while minimizing information loss. It accomplishes this by producing fresh, uncorrelated variables that maximize variance one after the other. As a result, PCA is an adaptive data analysis approach. Finding these new variables, the principal components, simplifies to solving a linear issue, and the new variables are specified by the dataset at present, not a priori. [15]

After the pre-processing was completed, a training set and a testing set were created from the pre-processed data. Finally, supervised (SVM, Random Forest) and unsupervised machine learning algorithms were applied on each of the datasets for the purpose of predicting the possibility of a DDoS attack on the basis of protocols. The classification of whether the traffic was benign or malicious was the result of the prediction. The following evaluation criteria were used to gauge how well the suggested system performed: confusion matrix, accuracy, precision and recall.

3.1 Data Selection

There are loads of DDoS attacks that an attacker can perform to harm the system. Some of them are already known to the Intrusion Detection Systems (IDS) of the organization while some are left undetected due to the way it is tailored to attack the system which make it look like normal traffic from the inside. Here, we are focusing on volume-based attacks that is, UDP and ICMP traffic and protocol attack which is Syn flooding.

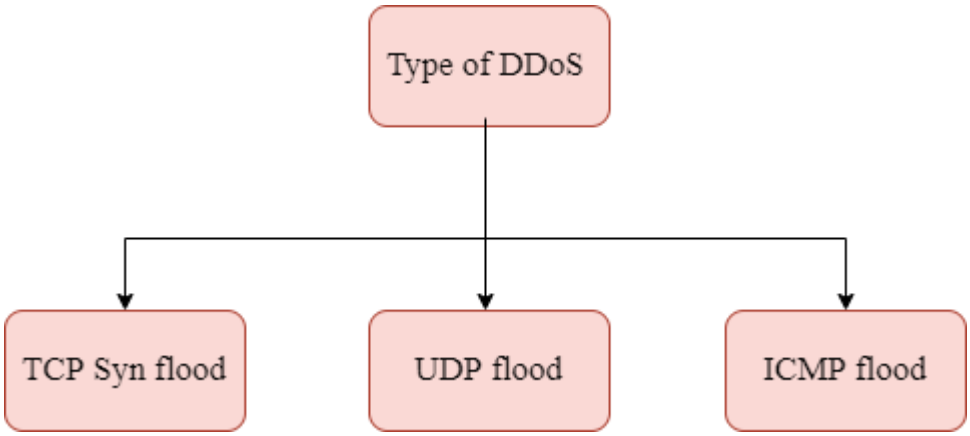

Fig.1 DDoS attack types

The amount of data or values of each protocol's dataset is as shown in table 1

DATASET	PROTOCOL	NO. OF ROWS
dataset_tcp.csv	TCP	29436
dataset_udp.csv	UDP	33587
dataset_icmp.csv	ICMP	41321

Table 1

3.2 Data Pre-processing

We are using SDN DDOS attack Dataset for our research. First of all, we preprocessed the dataset in order to create train and testing dataset. We performed cleaning of dataset by removing repetitive values and null values. Then we created train and test data frames from them. Inside each of the dataset, we have two classes, one for the attack and other depicting no attack. The dataset will then be divided into their respective classes/labels as given below:

- 0 – No Attack
- 1 – Attack

After successful formation of these datasets, I performed principal component analysis (PCA) on them and reduced the number of features for training. Then these final datasets are ready for ML trainings.

3.3 Extracting Features

The feature extraction was done by implementing Principal Component Analysis on the three datasets. Applying PCA ensures that the features that are reduced are uncorrelated to one another. To keep the variance, the value of the no. of components parameter is set to .93 to .97 to choose the fewest principal components.
Each dataset had a total of 19 features which were reduced as follows:

For the TCP dataset, the features were reduced to 11 from 19. Below shown is the scree plot for TCP

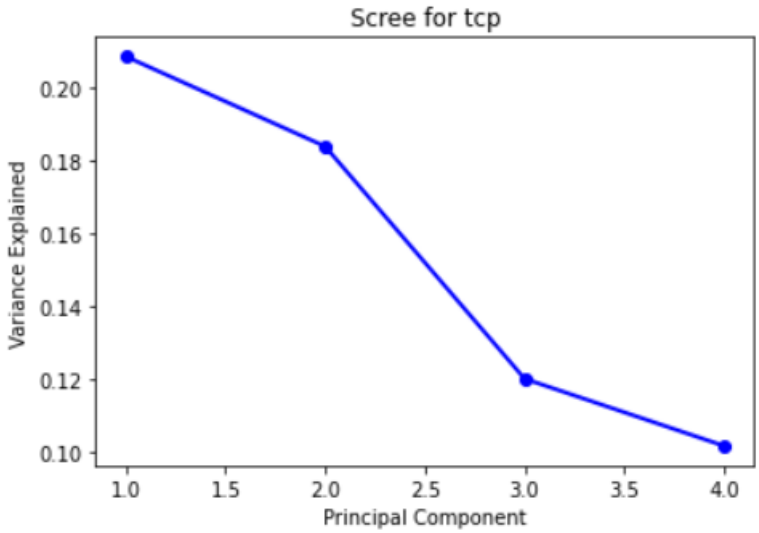

Fig.2 TCP Scree Plot

The UDP dataset originally consisted of 19 features that were brought down to 12 as well.

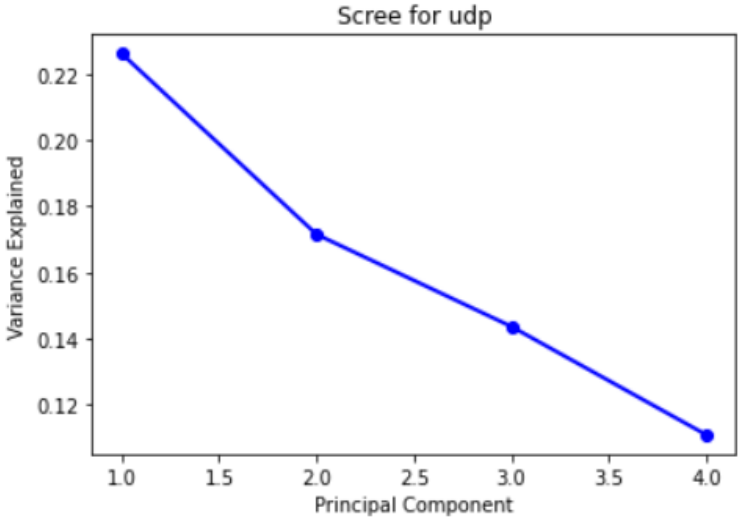

Fig.3 UDP Scree Plot

The ICMP dataset which had 19 features as well were reduced to 12 after the PCA.

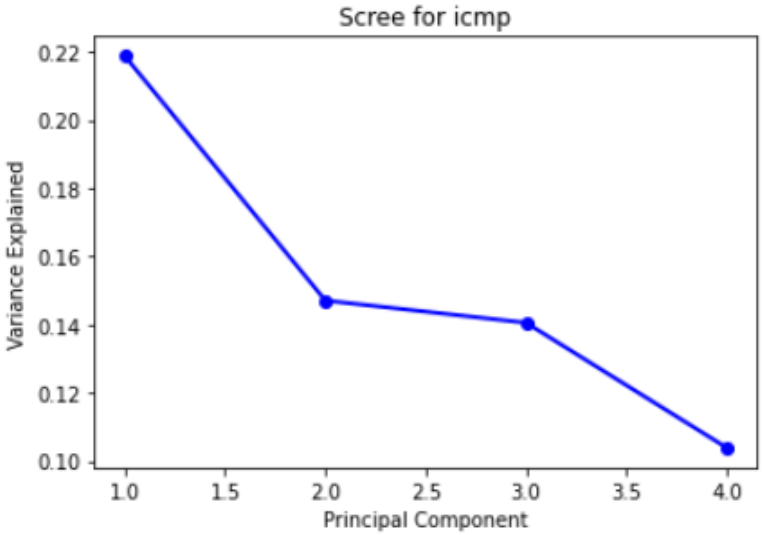

Fig.4 ICMP Scree Plot

4 Design Specification

This section covers the specifications of the projects and illustrates the development of the models. The original dataset used for the models is a CSV file containing DDoS attack traffic based on three different protocols (TCP, UDP, ICMP). To better check the efficiency and performance of our models, the original dataset was split into three distinct ones based on above mentioned three protocols. The newly created datasets were CSV files named dataset_tcp.csv, dataset_udp.csv and dataset_icmp.csv respectively.

Principal Component Analysis is then applied on each of the dataset with the goal of capturing new, more focused collection of characteristics that retains the majority of the important data. Simply put, PCA creates important information from raw data, or features, so that it may be used by models of machine learning to accomplish their objectives by reformatting, merging, and changing key features into new ones.[16]

Unsupervised classification technique K-Means, also known as clusterization, divides items into K groups according to their properties. The process of grouping involves minimizing the total distances between each object and the centroid of the group or cluster. The quadratic or euclidean distance is the common measurement. For our research, we have two clusters, DDoS attack and not DDoS attack. In SVM, our data are transformed using a method known as the kernel trick, and based on these modifications, it determines the best output boundary. To transform the multi-dimensional array into a continuous flattened one, the ravel() method is put to use here. The kernel parameter is set to linear so that linearly separated data can be classified. The performance and efficiency of Random Forest depends on the no. of trees because that's where the average prediction comes from.

The figure 2 shows the whole architecture of the models as well as how the datasets were converted and used for each model.

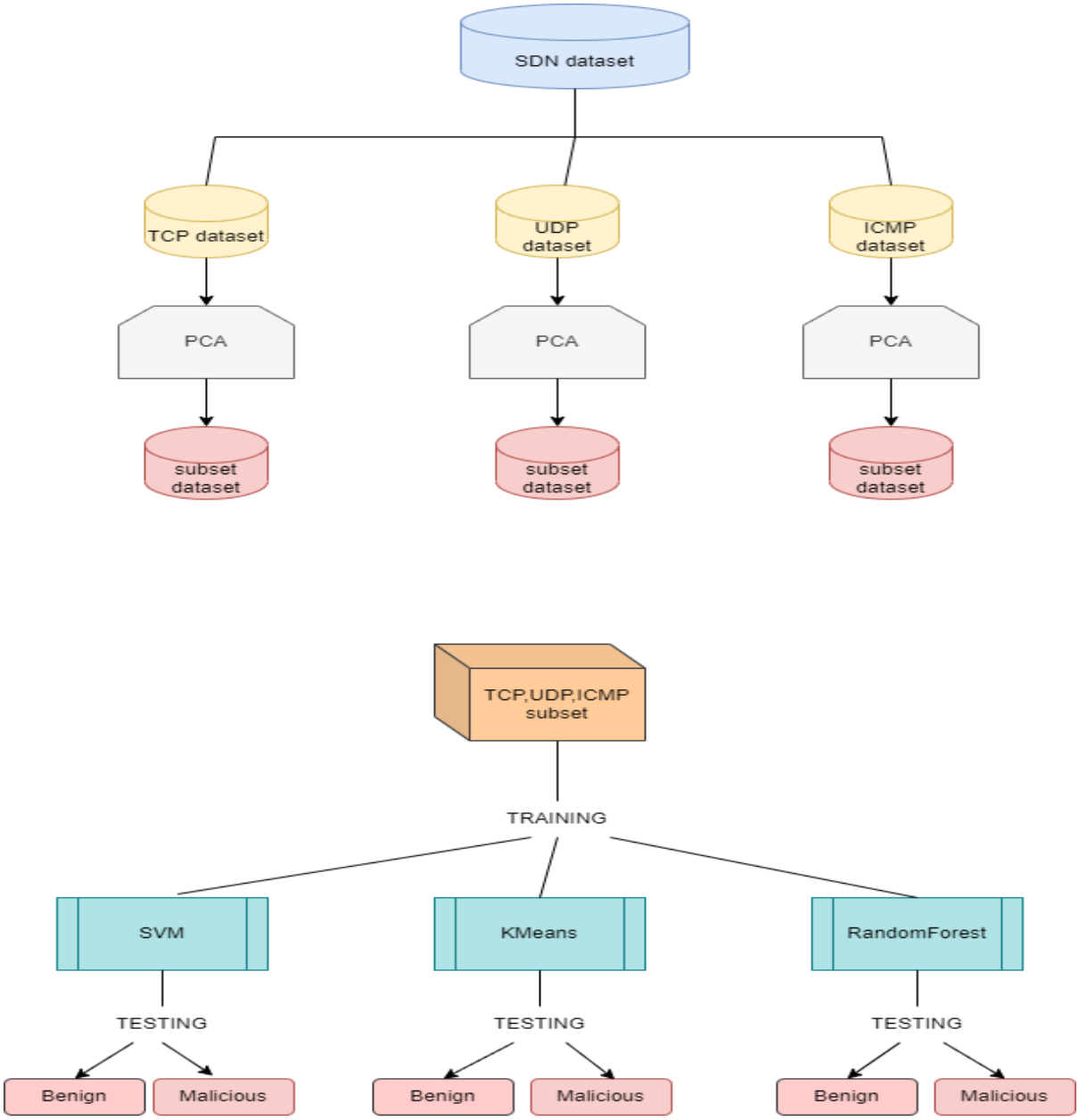

Fig.5 Working

5 Implementation

5.1 System

Machine learning requires the host machine to do some heavy lifting. For that reason, it's important to have a machine with the necessary hardware configurations capable of completing such tasks. The minimum requirements for a system are as follows:

- CPU: Intel i5 6th Generation Processor (2.4 GHz)
- RAM: 8GB (16GB recommended)
- Storage space: 15GB free space HDD or SSD

5.2 Machine

Your system should have a good, reliable internet connection for the initiation of the source code as well as the project. Below are the additional requirements:

- MS Excel – To analyse datasets
- Web Browser – Chrome/Firefox

5.3 Software Applications

- Anaconda Navigator – 64bits
- Python – Version 3 (recommended)

5.4 Packages required

Programming Language: Python
The following code was developed on Jupyter notebook. The packages that were needed for building the models are as mentioned below:

- Matplotlib 3.4.3
- Numpy 1.20.3
- Pandas 1.3.4
- Scikit-learn 0.24

6 Evaluation

To evaluate the models, test scenarios have been created and documented to test the generated models of SVM, Random Forest and K-means. The produced results are as shown below:

6.1 SVM model testing with TCP DDoS traffic

Training: The training of this model is done with TCP Syn flood DDoS attack traffic
Scenario: Checking the performance and accuracy of the model with TCP traffic
Dataset Used: dataset_tcp.csv
Outcome: The accuracy of the SVM model for TCP DDoS attack type is 75%

```
print("Accuracy:",metrics.accuracy_score(tcp_test_y.ravel(), tcp_pred_y))
```

```
Accuracy: 0.750832144555397
Confusion Matrix:
[[1518  709]
 [ 339 1640]]
tn, fp, fn, tp
1518 709 339 1640
```

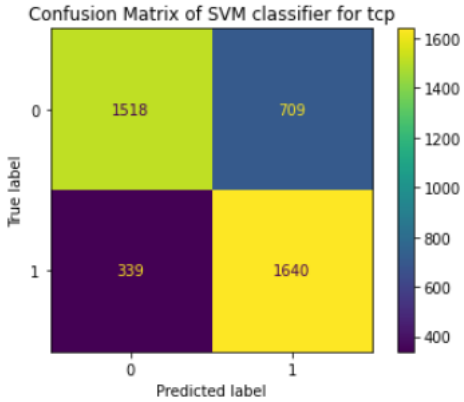

Fig. 6 SVM TCP TS

	precision	recall	f1-score	support
0	0.82	0.68	0.74	2227
1	0.70	0.83	0.76	1979
accuracy			0.75	4206
macro avg	0.76	0.76	0.75	4206
weighted avg	0.76	0.75	0.75	4206

6.2 SVM model testing with UDP DDoS traffic

Training: The training of this model is done with UDP flood DDoS attack traffic
Scenario: Checking the performance and accuracy of the model with UDP traffic
Dataset Used: dataset_udp.csv
Outcome: The accuracy of the SVM model for UDP DDoS attack type is 98.85%

```
print("Accuracy:",metrics.accuracy_score(udp_test_y.ravel(), udp_pred_y))
```

```
Accuracy: 0.988573846804909
Confusion Matrix:
[[2237   42]
 [  12 2435]]
tn, fp, fn, tp
2237 42 12 2435
```

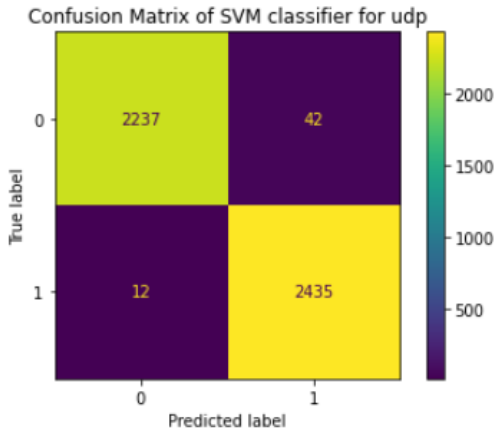

Fig.7 SVM UDP TS

	precision	recall	f1-score	support
0	0.99	0.98	0.99	2279
1	0.98	1.00	0.99	2447
accuracy			0.99	4726
macro avg	0.99	0.99	0.99	4726
weighted avg	0.99	0.99	0.99	4726

6.3 SVM model testing with ICMP DDoS traffic

Training: The training of this model is done with ICMP DDoS attack traffic
Scenario: Checking the performance and accuracy of the model with ICMP traffic
Dataset Used: dataset_icmp.csv
Outcome: The accuracy of the SVM model for ICMP DDoS attack type is 76.58%

```
print("Accuracy:",metrics.accuracy_score(icmp_test_y.ravel(), icmp_pred_y))
```

```
Accuracy: 0.7658817550398103
Confusion Matrix:
[[4521    0]
 [1382    0]]
tn, fp, fn, tp
4521 0 1382 0
```

	precision	recall	f1-score	support
0	0.77	1.00	0.87	4521
1	0.00	0.00	0.00	1382
accuracy			0.77	5903
macro avg	0.38	0.50	0.43	5903
weighted avg	0.59	0.77	0.66	5903

6.4 Random Forest model testing with TCP DDoS traffic

Training: The training of this model is done with TCP DDoS attack traffic
Scenario: Checking the performance and accuracy of the model with TCP traffic
Dataset Used: dataset_tcp.csv
Outcome: The accuracy of the Random Forest classifier model for TCP DDoS attack type is 94.34%

```
print("Accuracy:",metrics.accuracy_score(tcp_test_y.ravel(), tcp_pred_y))
```

```
Accuracy: 0.9434141702330004
Confusion Matrix:
[[4506   15]
 [1382    0]]
tn, fp, fn, tp
4506 15 1382 0
```

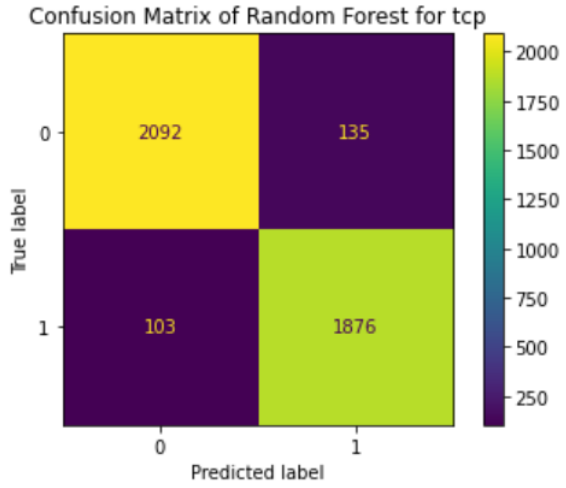

Fig.9 Random Forest TCP

6.5 Random Forest model testing with UDP DDoS traffic

Training: The training of this model is done with UDP DDoS attack traffic
Scenario: Checking the performance and accuracy of the model with UDP traffic
Dataset Used: dataset_udp.csv
Outcome: The accuracy of the Random Forest classifier model for UDP DDoS attack type is 99.72%

```
print("Accuracy:",metrics.accuracy_score(udp_test_y.ravel(), udp_pred_y))
```

```
Accuracy: 0.9972492594159966
Confusion Matrix:
[[4506   15]
 [1382    0]]
tn, fp, fn, tp
4506 15 1382 0
```

	precision	recall	f1-score	support
0	1.00	0.99	1.00	2279
1	1.00	1.00	1.00	2447
accuracy			1.00	4726
macro avg	1.00	1.00	1.00	4726
weighted avg	1.00	1.00	1.00	4726

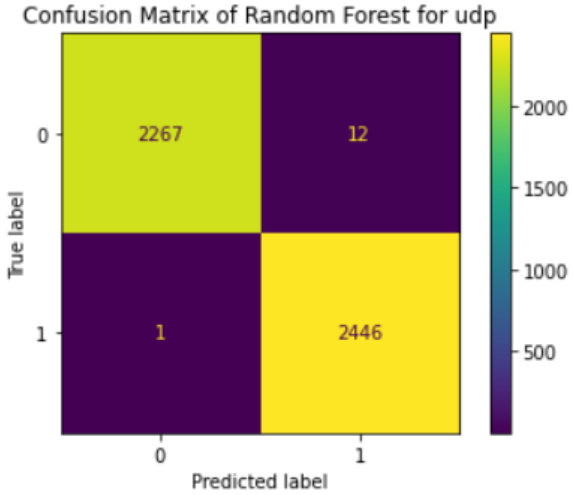

Fig.10 Random Forest UDP

6.6 Random Forest classifier model testing with ICMP DDoS traffic

Training: The training of this model is done with ICMP DDoS attack traffic
Scenario: Checking the performance and accuracy of the model with ICMP traffic
Dataset Used: dataset_icmp.csv
Outcome: The accuracy of the Random Forest classifier model for ICMP DDoS attack type is 76.33%

```
print("Accuracy:",metrics.accuracy_score(icmp_test_y.ravel(), icmp_pred_y))
```

```
Accuracy: 0.7633406742334407
Confusion Matrix:
[[4506   15]
 [1382    0]]
tn, fp, fn, tp
4506 15 1382 0
```

	precision	recall	f1-score	support
0	0.77	1.00	0.87	4521
1	0.00	0.00	0.00	1382
accuracy			0.76	5903
macro avg	0.38	0.50	0.43	5903
weighted avg	0.59	0.76	0.66	5903

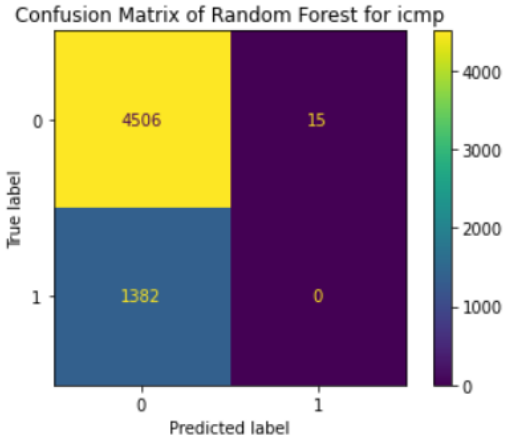

Fig.12 Random Forest ICMP

6.7 K-means classifier model testing with TCP DDoS traffic

Training: The training of this model is done with TCP DDoS attack traffic
Scenario: Checking the performance and accuracy of the model with TCP traffic
Dataset Used: dataset_tcp.csv
Outcome: The accuracy of the K-means classifier model for TCP DDoS attack type is 58%

```
print(confusion_matrix(tcp['Cluster'], myKMC.labels_))
```

```
[[9468 6102]
 [6279 7587]]
```

	precision	recall	f1-score	support
0	0.60	0.61	0.60	15570
1	0.55	0.55	0.55	13866
accuracy			0.58	29436
macro avg	0.58	0.58	0.58	29436
weighted avg	0.58	0.58	0.58	29436

6.8 K-means classifier model testing with UDP DDoS traffic

Training: The training of this model is done UDP attack traffic
Scenario: Checking the performance and accuracy of the model with UDP traffic
Dataset Used: dataset_udp.csv
Outcome: The accuracy of the K-means classifier model for UDP DDoS attack type is 62%

```
print(classification_report(udp['Cluster'], myKMC.labels_))
```

```
[[10865  4998]
 [ 7561  9658]]
```

	precision	recall	f1-score	support
0	0.59	0.68	0.63	15863
1	0.66	0.56	0.61	17219
accuracy			0.62	33082
macro avg	0.62	0.62	0.62	33082
weighted avg	0.63	0.62	0.62	33082

6.9 K-means classifier model testing with ICMP DDoS traffic

Training: The training of this model is done with ICMP DDoS attack traffic
Scenario: Checking the performance and accuracy of the model with ICMP traffic
Dataset Used: dataset_icmp.csv
Outcome: The accuracy of the K-means classifier model for ICMP DDoS attack type is 50%

	precision	recall	f1-score	support
0	0.72	0.58	0.64	31902
1	0.14	0.23	0.17	9419
accuracy			0.50	41321
macro avg	0.43	0.40	0.41	41321
weighted avg	0.59	0.50	0.53	41321

```
print(confusion_matrix(icmp['Cluster'], myKMC.labels_))
```

```
[[18465 13437]
 [ 7258  2161]]
```

6.10 Discussion

Upon conducting the experiments with different models and three different datasets, we came to realize that our datasets work best with SVM as also proven in other studies as well and the worst with K-means, the reason being this algorithm works efficiently with unlabelled data. The highest accuracy that we got was from the SVM model on the UDP DDoS dataset that is of 98.8% which justifies that SVM works better in detection such types of attacks. The least accuracy was observed of the K-means classifier on our datasets.

Test Scenarios	Model	Accuracy
TS1	SVM model testing with TCP DDoS traffic	75%
TS2	SVM model testing with UDP DDoS traffic	98.85%
TS3	SVM model testing with ICMP DDoS traffic	76.58%
TS4	Random Forest model testing with TCP DDoS traffic	94.34%
TS5	Random Forest model testing with UDP DDoS traffic	99.72%
TS6	Random Forest model testing with ICMP DDoS traffic	76.33%
TS7	K-means classifier model testing with TCP DDoS traffic	58%
TS8	K-means classifier model testing with UDP DDoS traffic	62%
TS9	K-means classifier model testing with ICMP DDoS traffic	50%

Table 2 Summary table

7 Conclusion and Future Work

The overall research depicts the comparison of supervised and unsupervised algorithms for distinguishing the DDoS attack traffic as malicious and benign data. The presented models show that the maximum accuracy is achieved by Random Forest classifier with 99.7% accuracy on the UDP dataset and 94.74% accuracy on the TCP testing dataset. The result of K-means is not that great with these datasets with accuracies as low as 36.63% and 58.61% on the UDP and ICMP testing datasets respectively.

Due to hardware limitations as well as constraints on time, we were incapable of capturing real-time data and creating a dataset of our own for utilizing in this research. Also, we focused on DDoS attacks on only three protocols. In the future with more time and software/hardware capabilities, we'd like to test and improve the accuracy of the model by creating more datasets specific to protocols other than the ones mentioned in this research.

8 Acknowledgement

I would like to express my sincere gratitude to my supervisor Prof. Michael Pantridge for his continued support throughout my third semester for my Masters' thesis research. I would like to give my special thanks to him for his patience, motivation and immense knowledge which helped me greatly. His guidance really made a lot of difference in my research. I couldn't have imagined a better mentor for my thesis study.

9 References

[1] Buczak, A.L. and Guven, E. (2016) A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys & Tutorials, 18, 1153-1176.

[2] Ahuja, Nisha; Singal, Gaurav; Mukhopadhyay, Debajyoti (2020), "DDOS attack SDN Dataset", Mendeley Data, V1, doi: 10.17632/jxpfjc64kr.1

[3] KDD Cup 1999 Data, 1999

[4] 1999 DARPA Intrusion Detection Evaluation Dataset | MIT Lincoln Laboratory, 1999

[5] Li, D., Yu, C., Zhou, Q. and Yu, J., 2018. Using SVM to Detect DDoS Attack in SDN Network. *IOP Conference Series: Materials Science and Engineering*, 466, p.012003.

[6] Khuphiran P, Leelaprute P, Uthayopas P, Ichikawa K and Watanakeesuntorn W (2018), "Performance Comparison of Machine Learning Models for DDoS Attacks Detection" Information on Computer Science and Engineering Conference IEEE pp. 1-4

[7] Medium(2019), "Intrusion detection in KDD99 dataset using machine learning"

[8] Y. Chen, J. Hou, Q. Li and H. Long, "DDoS Attack Detection Based on Random Forest," 2020 IEEE International Conference on Progress in Informatics and Computing (PIC), 2020, pp. 328-334, doi: 10.1109/PIC50277.2020.9350788.

[9] Najar, A. and Manohar Naik, S., 2019. DDoS attack detection using MLP and Random Forest Algorithms. *International Journal of Information Technology*, 14(5), pp.2317-2327.

[10] Agrawal, S. and Rajput, R., 2017. *Denial of Services Attack Detection using Random Forest Classifier with Information Gain*. IJEDR, 2321-9399

[11] Yusof, A., Udzir, N. and Selamat, A., 2016. An Evaluation on KNN-SVM Algorithm for Detection and Prediction of DDoS Attack. *Trends in Applied Knowledge-Based Systems and Data Science*, pp.95-102.

[12] J. Cui, J. Zhang, J. He, H. Zhong and Y. Lu, "DDoS detection and defense mechanism for SDN controllers with K-Means," 2020 IEEE/ACM 13th International Conference on Utility and Cloud Computing (UCC), 2020, pp. 394-401, doi: 10.1109/UCC48980.2020.00062.

[13] Umarani S and Sharmila D (2014), "Predicting application layer DDoS attacks using machine learning algorithms". International Journal of Computer, control Quantum and information Engineering 8.10

[14] Gal, M. and Rubinfeld, D., 2019. *Data Standardization*. New York University Law Review, 0028-7881

[15] Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences(2020), "Principal component analysis: a review and recent developments "

[16] Schlens, J., 2014. *A Tutorial on Principal Component Analysis* 1404.1100