Evaluating the Efficacy of Machine Learning in Detecting DoS & DDoS Attacks: A Comprehensive Dataset Analysis

Munirah Ab Rahman1[1]*, Mohd Azuwan Efendy Mail[1] & Mohd Faizal Ab Razak[2]

[1]Department of Information Technology and Communication, Politeknik Mersing
[2]Faculty of Computing, Universiti Malaysia Pahang
*Corresponding author: munirah@tvet.pmj.edu.my

Abstract
Machine learning is currently being widely employed to create malware detection systems (MDS). These systems can detect and categorize various types of cybercrime, including DoS and DdoS attacks. These attacks may involve multiple unique IP addresses and machines, and can be facilitated by malware. DoS and DDoS attack scans are carried out on a daily basis, and non-profit multinational companies, regardless of size, can fall victim to them. Such attacks can cause a significant slowdown or even bring down the online services, email, websites, and other digital operations of these companies. Cybersecurity operations may sometimes be disrupted by DoS and DDoS attacks, which can allow criminal activities such as data theft and network infiltration to take place, resulting in the loss of valuable company data. Our DoS and DDoS dataset for the malware detection system employs a single methodology, which involves the use of Python code. This paper will focus on assessing the accuracy of the DoS and DDoS dataset within the malware detection system.

Keywords: Dataset; Dos & Ddos; Machine Learning; Malware; Malware Detection System.

1. Introduction

Nowadays, Cyber-attacks are everywhere. Many of the attacks are based on their own goals or any malicious entities who wanted to aim to disrupt the system of the service of the specific company. There are many types of attacks when some of the attacker's identity remains hidden by using legitimate third-party components. The attacker of DoS and DDoS can set the victim's IP address as their desired target IP address and transfer it into packets to reflector servers. There are many cyber-attacks such as Dos and DDoS attacks that are usually found by humanly instructed systems that consist of different devices with internet access. The existence of bots can perform many types of attacks such as DoS and DDoS when a computer is infected by malware with specific software (Idika, 2007) DoS & DDoS attack will easily affect the client side of the system.

Machine learning is used to detect the DoS and DDoS. Machine learning can classify classes within a defined data set. Classification is to classify into two classes which is benign and malware. Malware can be referred to by many names. B. Malicious Software or Malicious Code. Malware affects the world as we know it. In 1988, an increase in incidents in cybersecurity systems indicated that malware was prevalent. Research on knowledge of malware functionality indicates that malware detection is an area of great interest. It should be studied not only for the community, but also for the general public (Eduardo, 2022). A malware developer researcher realized the implementation of a malware detector. This report aims to investigate and discover the accuracy of DoS and DDoS techniques for malware detection systems.

2.0 Literature Review

2.1 DDoS Detection using Machine Learning Techniques

In this article, Eduardo Alexandre Romao Coelho describes network-based services that improve business flexibility and scalability. This article made a distinction between DoS and DDoS attacks. This article discusses various types of DoS attacks, including their ease of detection, attack speed, traffic volume, execution type, and source tracking. Additionally, DDoS attacks are examined, focusing on UDP floods, ICMP floods, SYN floods, HTTP floods, and Smurf attacks. The methodology in this article looks for the most accurate attack detection method with a low false positive rate and a high directed positive rate by using 4 classifiers which is decision trees, Naive Bayes, random forests, and MLPs (neural networks) work (Eduardo, 2022).

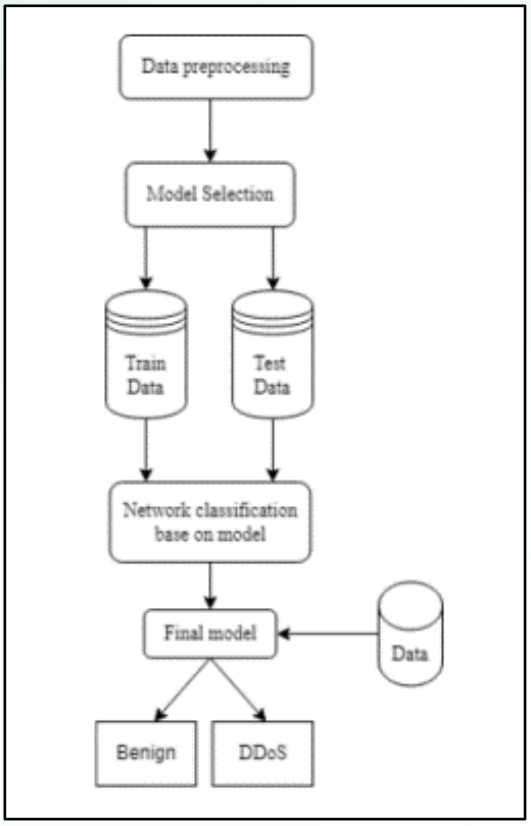


Figure 1: The proposed method to be implemented (Eduardo, 2022)

For the calculation to the access of the model's performance, the alculation is defined as:

1. Accuracy – the proportion of data instances classified correctly by a model

$$Precision(P) = \frac{TP}{TP + FP}$$

2. Precision – The percentage of correctly classified attacks divided by the total predicted attacks.

$$Accuracy(A) = \frac{TPR + TNR}{TPR + TNR + FPR + FNR}$$

3. Recall – The proportion of correctly classified attacks in relation to the total number

$$Recall(R) = \frac{TP}{TP + FN}$$

4. F1-score – is a measure of the accuracy of a model on a dataset. It is the precision and recall harmonies

$$F1 - score(F) = \frac{2PR}{P + R}$$

Various sizes of tensile and test samples were investigated, showing no significant effect in this case. Eduardo concludes that this can be attributed

to a carefully selected set of features and well-tuned models (Eduardo, 2022).

2.2 Malware Detection Using Honeypot and Machine Learning

This article explores the growing number of malwares. In this article, Muhamad Malik Matin proposed an architecture for malware detection using honeypots, which are vulnerable systems to networks. (Muhamad, 2020) The motive of honeypots is to adapt the technique of an attacker's tools or methods. This article uses both decision algorithms and support vector machines (SVMs). The proposed architecture is designed to detect vulnerability based on its behavior, and the system can be trained (Muhamad, 2020)

The proposed architectural design includes network devices, routers, honeypots, data analytics, and the actual system. All network access traffic is directed into the network, and routers are also responsible for connecting external networks to the intranet and forwarding packets to honeypots. Finally, honeypots can capture and store traffic packets from the internal network. Collected packages are used to generate analysis proposals (Muhamad, 2020).

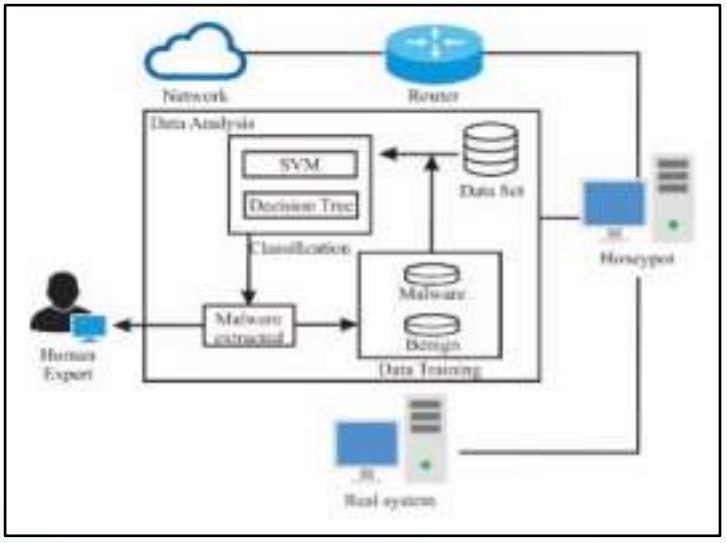


Figure 2: Proposed architecture (Muhamad, 2020)

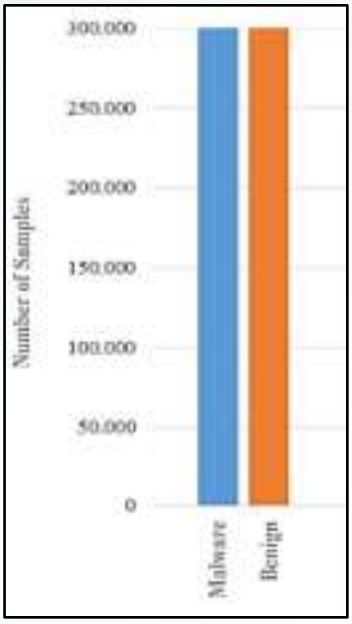


Figure 3: Distribution of malicious and benign samples dataset (Muhamad, 2020)

A percentage split test is used in this method. It is to divide the data into two parts, namely test data and training data, so that all data will be validated overall (Muhamad, 2020)

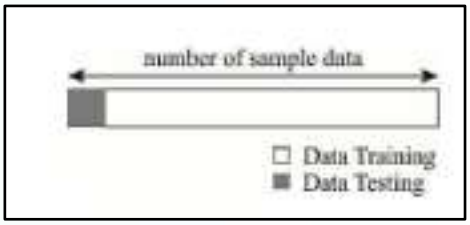


Figure 4: Distributed of dataset for testing data and training data (Muhamad, 2020)

This article employs four parameters for evaluation: True Positive Rate (TPR), False Positive Rate (FPR), Precision, Recall, and Accuracy:

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

$$Precission = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Finally, this malware detection architecture based on honeypots and machine learning has been presented. The classification in this study uses the Support Vector Machine (SVM) and Decision Tree algorithms, yielding high accuracy and efficiency. The Validation test is determined by ten experiments. Supervised machine learning with labelled data was used in this study. In the following stage, the proposed design (Muhamad, 2020).

2.3 DDoS Attack Detection using Machine Learning Techniques in Cloud Computing Environments

Marwane Zekri, Said El Kafhali, Noureddine Aboutabit, and Youssef Saadi contributed to this article. They wanted to create a DDoS detection system that could mitigate the DDoS threat using the C.4.5 algorithm (Zekri, 2017). The algorithm creates a decision tree that performs automatic, efficient signature identification for DDoS flooding assaults when paired with signature detection approaches. (Zekri, 2017). Other machine learning methods will be contrasted using the outcome. (Zekri, 2017).

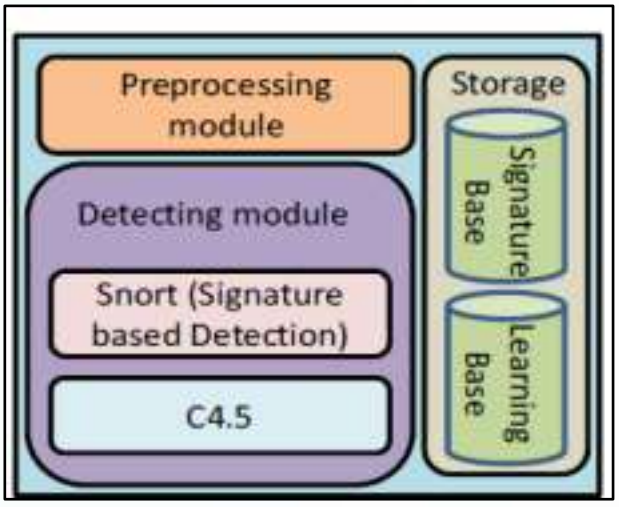


Figure 5: Preprocessing Engine captures packets in a specific format by removing redundant information (Zekri, 2017)

According to the diagram, the most important engine, the preprocessing engine, captures packets in a specific format by removing redundant information that has little relevance for detection. For example, signature-based detection efficiently detects known attacks by matching specific network events to rules stored in a knowledge base. This technique collects data related to legitimate users over a period of time and applies machine learning to that data. To allow us to determine whether a particular user is legitimate.

To overcome the overfitting problem, we use the C4.5 algorithms to build decision trees that select attributes as splitting criteria based on entropy-based boosting ratios. In the decision tree, the attribute with the highest winning probability is chosen as the splitting criterion. Then it is split into several subsets and the splitting procedure is repeated until all the data in the subsets are in the same class or all gain ratios are equal (Zekri, 2017).

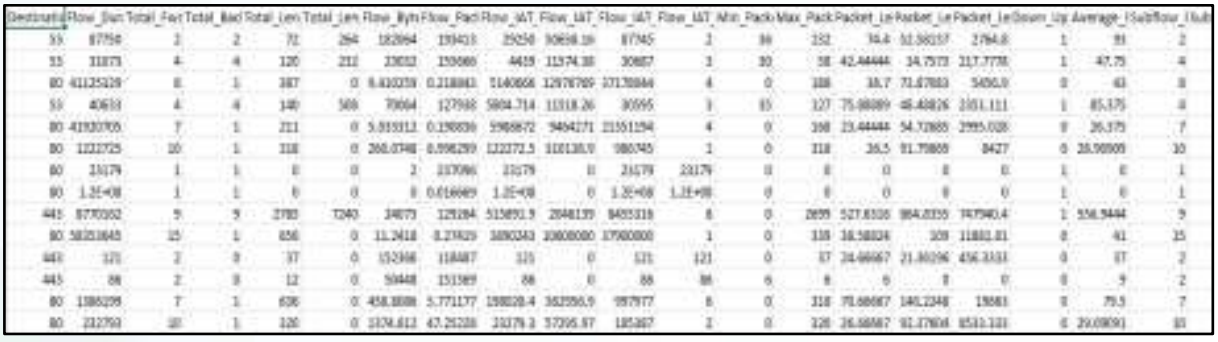
3. Methodology

3.1 Jupyter Notebook

The methodology involves using Jupyter Notebook to develop a system for calculating the accuracy of the dataset. Jupyter Notebook permits users to compile all aspects of an information project in one place creating it easier to indicate the whole method of a project to meant audience. Through the web-based application, users will produce information visualizations and alternative parts of a project to share with others vis the platform.

However, for the sample dataset, we tend to use Kaggle website, to urge the small print of the dataset. Kaggle is an internet community platform for knowledge scientists and machine learning enthusiasts. Kaggle permits users to collaborate with alternative users. realize and publish dataset use GPU integrated notebooks, and vie with alternative knowledge scientists to resolve knowledge science challenges.

For this assignment, we tend to use dataset from (URL:https://www.kaggle.com/code/hamzasamiullah/ml-analysis-application-layer-dosattack-dataset/data). This dataset is concerning Machine Learning analysis: Application Layer DDoS & DoS attack. Figure 9 shows that the sample dataset that we tend to gain from the Kaggle.



Destination	Flow_Dur	Total_Fwd	Total_Bwd	Total_Len	Total_Len	Flow_Byts	Flow_Pack	Flow_IAT_	Flow_IAT_	Flow_IAT_	Flow_IAT_	Min_Packe	Max_Packe	Packet_Le	Packet_Le	Packet_Le	Down_Up_	Average_P	SubFlow_F	Sub
53	87750	2	2	72	264	182994	159413	29250	30858.19	87745	2	20	252	36.4	52.58157	2764.8	1	88	2	
53	31873	4	4	120	232	23032	150580	4429	11374.38	30857	3	20	58	42.44444	34.7573	217.7778	1	47.75	4	
80	41125329	8	1	167	0	8.610259	0.218843	5140666	12976789	17170844	4	0	188	38.7	71.67883	5406.9	0	63	8	
53	40633	4	4	140	568	70664	127948	5804.714	11318.26	30595	3	85	127	75.88889	48.40826	2151.111	1	85.375	4	
80	41930705	7	1	211	0	5.015312	0.190836	5986872	9464171	21551254	4	0	168	23.44444	54.72865	2995.028	0	26.375	7	
80	1222725	10	1	318	0	260.0748	0.998259	122272.5	318138.9	980745	1	0	318	26.5	91.79869	8427	0	26.90909	10	
80	21179	1	1	0	0	2	137086	21179	0	21179	21179	0	0	0	0	0	1	0	1	
80	1.2E+08	1	1	0	0	0	0.016089	1.2E+08	0	1.2E+08	1.2E+08	0	0	0	0	0	1	0	1	
443	8770162	9	9	2783	7240	24873	125284	515891.9	2648139	8455318	8	0	2699	527.8316	984.0359	747940.4	1	556.9444	9	
80	58353845	15	1	656	0	11.2418	0.27429	3890343	10808080	17900000	1	0	338	38.58824	109	11881.81	0	41	15	
443	121	2	0	37	0	152066	116487	121	0	121	121	0	37	26.66667	21.86296	456.3333	0	37	2	
443	88	2	0	12	0	59448	151589	88	0	88	88	6	6	6	0	0	0	9	2	
80	1386299	7	1	636	0	458.8888	5.771177	198028.4	362556.9	957977	6	0	318	70.66667	140.2248	19881	0	79.5	7	
80	212750	10	1	320	0	1374.612	47.26228	23179.3	57205.97	185267	1	0	320	26.66667	91.27604	8533.333	0	29.09091	10	

Figure 9: Sample dataset

It has seventy-eight columns in total during this dataset, however solely ten columns square measure employed in this assignment. This internet sites give varies kinds of datasets, and its free resources for others.

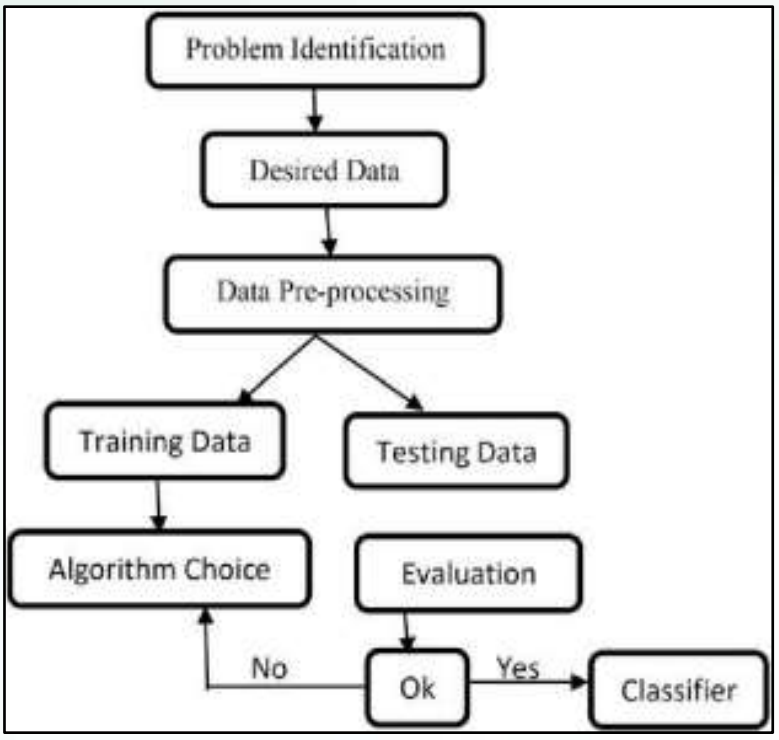


Figure 10: The Method Model on However the Dataset Are Acknowledge

Figure 10 shows the method model on however the dataset are acknowledge and also the flow on input is method via stages till seem the result. To implement this algorithmic rule, we tend to use python language in Jupyter Notebook to search out the accuracy of the chosen dataset.

For the python code, figure below will be displayed:

```
In [1]: import pandas as pd
        from sklearn.tree import DecisionTreeClassifier
        from sklearn.model_selection import train_test_split
        from sklearn.metrics import accuracy_score
        from sklearn.svm import SVC
        from sklearn import tree
```

Figure 11: Python Code

At first, we have to import all the necessary library for this assignment, as shown within the higher than figure, we tend to import pandas as pd as a short form for us if every time we want to use pandas, we tend to simply get to write pd. Next, we decide on the import call tree classifier for the model. After that, we tend to import train test split library to urge the prediction worth for the accuracy. Then, we tend to import accuracy scores, to calculate the accuracy of our dataset.

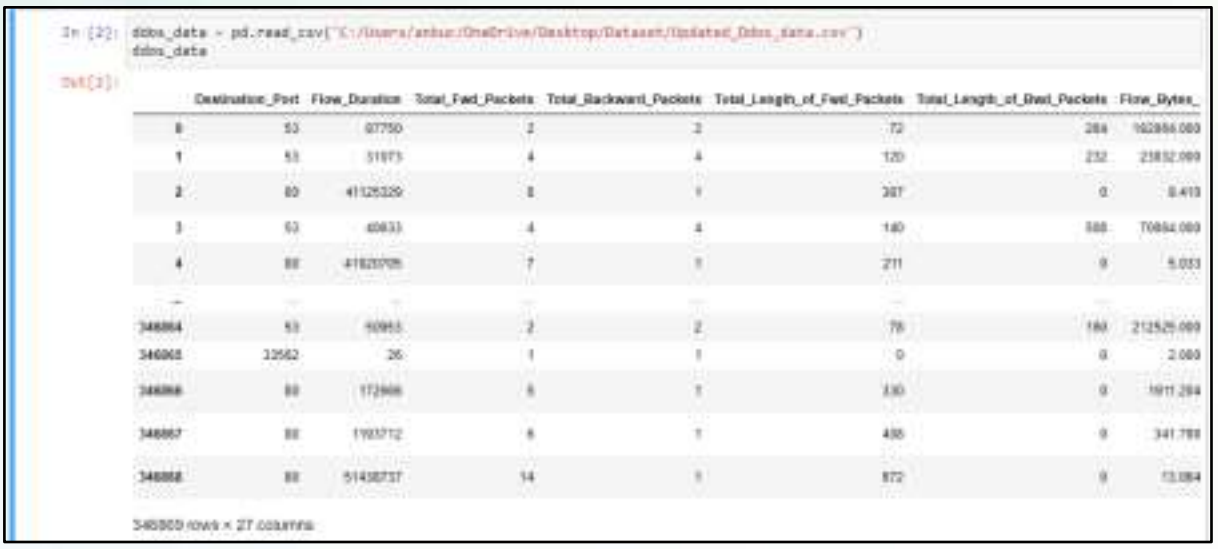
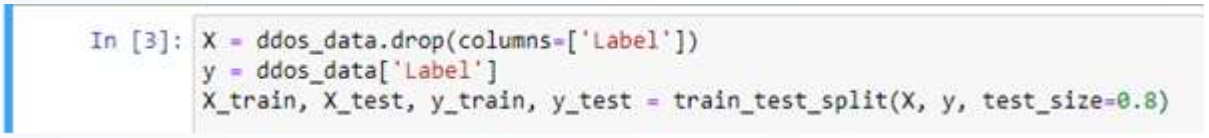


Figure 12: Dataset

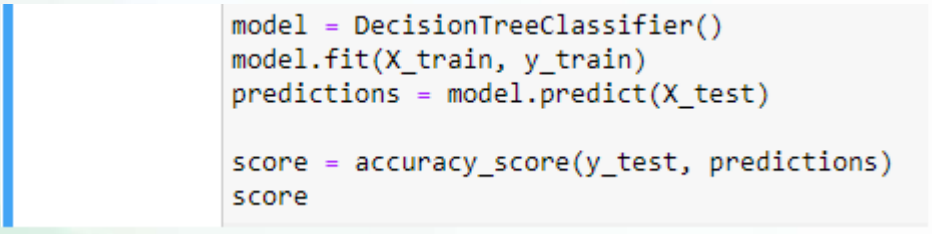
After imported all the required library, we need to import the {data|the info|the information} set within the Jupyter Notebook as for it to browse the data for subsequently code. Figure 12 shown the dataset once we import it in the Jupyter Notebook.



```
In [3]: X = ddos_data.drop(columns=['Label'])
        y = ddos_data['Label']
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.8)
```

Figure 13: Information 1

As for the part, we'd like to separate the info into 2 so the machine is ready to seek out the prediction of the result thus get the accuracy, so we tend to produce a brand-new information without the output for the machine to predict the end result. Then we name it as 'X' for the new dataset. Then, we'd like to feature another line where to call the output, so we need to use 'y' as the variable that may contain the worth of the output that is in this case, we use 'Label' as the outcome of this dataset.



```
model = DecisionTreeClassifier()
model.fit(X_train, y_train)
predictions = model.predict(X_test)

score = accuracy_score(y_test, predictions)
score
```

Figure 14: Information 2

At first, ought to produce a replacement model to urge the category of the information. Then, we need to create model work with the input set and the output set which is 'X' and 'y'. Next, we need to raise the model to do the prediction thus the prediction model be created. Finally, accuracy score is referred to as with the given arguments.

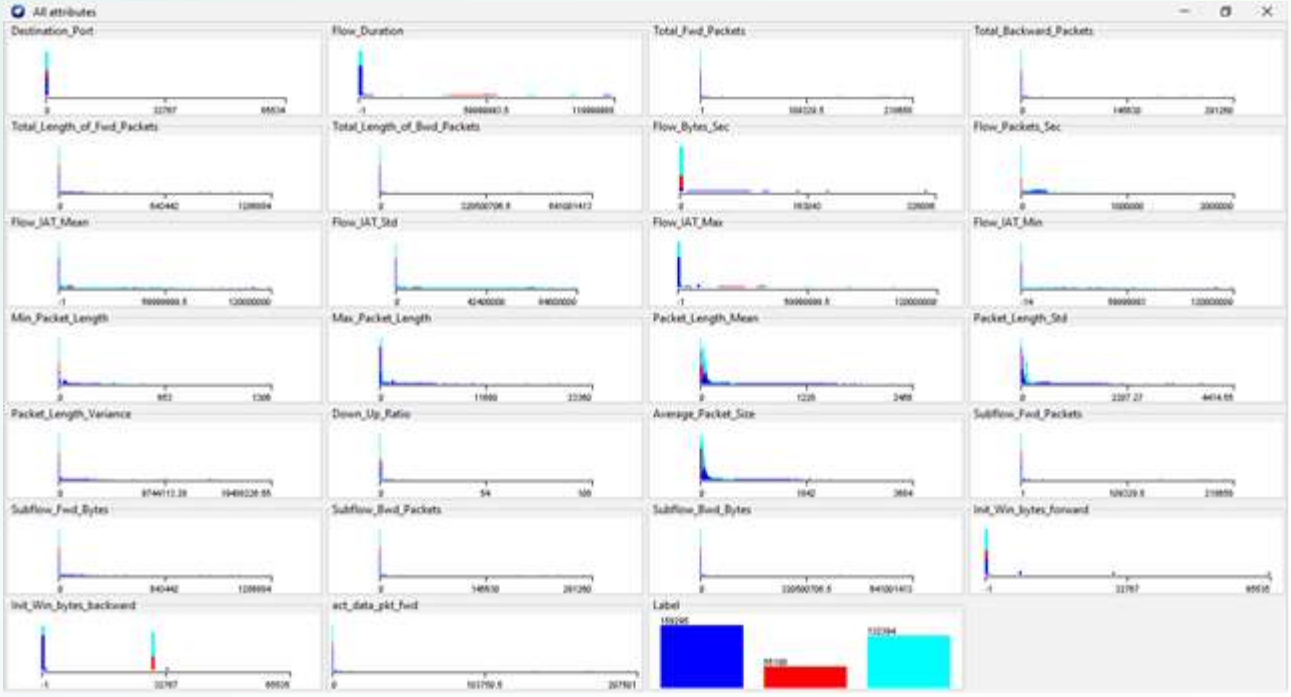


Figure 15: The visualization of all the attributes.

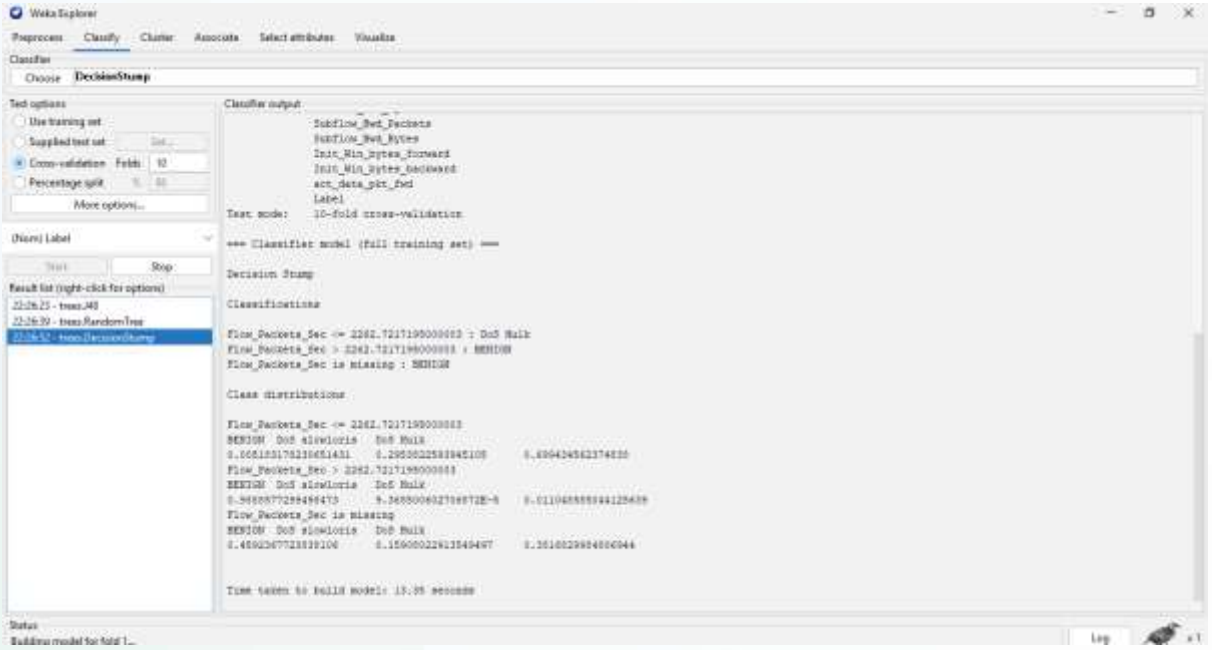


Figure 16: The class distributions of the dataset.

Figure 16 shows the class distributions of the dataset using the weka to analyze the data.

4. Results and Discussion

Figure 17 is the result that we would get after we run the previous code, as for test size = 0.8, we get the accuracy as 0.9998198192225906.



Figure 17: Result

Test Size	Accuracy
0.2	0.99995675613334217
0.4	0.999884683022458
0.6	0.9998030001633658

Table 1: The accuracy varies with the size of the test

The table above shows that the accuracy varies with the size of the test, with all four sizes providing very good accuracy (above 90%). As a result, we can conclude that this dataset is successful. It means that if the model predicts a positive value, what are the chances that the model is correct.



Figure 18: DDoS-recommender.dot

As for this, we write a code to get file name 'ddos-recommender.dot' to get the prediction in graphical shape to get the exact view how can the machine get the predictions.

5. Conclusion

A DDoS & DoS attack is venturesome to network security. it's a threat to the business moreover as applications which will solely grow within the future. As a result, it's up to all or any these corporations and platforms to make sure their safety and to continue trying to stop and detecting these varieties of attacks. Machine Learning was utilized in this project to develop a DDoS & DoS detection system. This goal may be to own been accomplished with success. The dataset used was massive, with seventy-eight options, despite the very fact that solely ten were chosen. decision Trees were used as classifiers. The investigation solely with numerous sizes for check and train samples discovered that it had no vital impact during this case. this might be attributed to a felicitous set of attributes and therefore the top quality of the mark model.

As for the longer-term work, the model must be compelled to be tested in an exceedingly real-world situation with live network traffic capture - differing

types of DDoS & DoS - and can be improved additional. Following these milestones, Associate in Nursing IDS supported the particular project may be created.

6. Acknowledgement

We would like to express our sincere gratitude to Polytechnic Mersing Johor and University Malaysia Pahang for their support and assistance throughout this project.

References

Agarwal, A., Singh, R., & Khari, M. (2022). Detection of DDOS Attack Using IDS Mechanism: A Review, in Proceedings of 2022 1st International Conference on Informatics, ICI 2022, 2022, pp. 36–46. https://doi. 10.1109/ICI53355.2022.9786899

Anna University and IEEE Aerospace and Electronic Systems Society, (2019) International Carnahan Conference on Security Technology (ICCST): ICCST 2019: IEEE 53rd International Carnahan Conference on Security Technology: October 01-03, 2019, Anna University, Chennai, India

Chen, Y., Ma, X., & Wu, X. (2013). DDoS detection algorithm based on preprocessing network traffic predicted method and chaos theory, IEEE Communications Letters, vol. 17, no. 5, pp. 1052–1054. https://doi. 10.1109/LCOMM.2013.031913.130066

Coelho, Eduardo. (2022). DDoS Detection using Machine Learning Techniques. https://www.researchgate.net/profile/EduardoCoelho23/publication/360246281_DDoS_Detection_using_Machine_Learning_Techniques/links/626afc86bfd24037e9 dbbf81/DDoS-Detection-using-Machine-Learning-Techniques.pdf

Dong, S., & Sarem, M. (2020). DDoS Attack Detection Method Based on Improved KNN with the Degree of DDoS Attack in Software-Defined Networks, IEEE Access, vol. 8, pp. 5039–5048. https://doi. 10.1109/ACCESS.2019.2963077

Gu, Y., Li, K., Guo, K., & Wang, Y. (2019). Semi-supervised k-means ddos detection method using hybrid feature selection algorithm, IEEE Access, vol. 7, pp. 64351–64365. https://doi.10.1109/ACCESS.2019.2917532

Hussain, F., Abbas, S. G., Husnain, M., Fayyaz, U. U., Shahzad, F., & Shah, G. A. (2020). IoT DoS and DDoS Attack Detection using ResNet, in Proceedings - 2020 23rd IEEE International Multi-Topic Conference, INMIC 2020. https://doi.10.1109/INMIC50486.2020.9318216

Idika, N., & Mathur, A. P. (2007). A Survey of Malware Detection Techniques.

Muhamad, I., Matin, M., & Rahardjo, B. (2020). Malware Detection Using Honeypot and Machine Learning. https://public.gdatasoftware.com/Presse/Publikationen/Malware_Rep

Vinayakumar- Alazab, R.M, Soman, K. P., Poornachandran, P., Al-Nemrat, A., & Venkatraman, S. (2019). Deep Learning Approach for Intelligent Intrusion Detection System," IEEE Access, vol. 7, pp. https://doi. 10.1109/ACCESS.2019.2895334

Zekri, M., Kafhali, S. el, Aboutabit, N., & Saadi, Y. (2017). DDoS Attack Detection using Machine Learning Techniques in Cloud Computing Environments. https://doi.10.1109/CloudTech.2017.8284731