DDOS Attack Detection Using Supervised Machine Learning Techniques

1st Aravinda Krishna Gorantla 2nd Rakesh Karnekanti 3rd Dharani Atukuri 4th Chandrakanth Bikumandla

***Abstract*—Websites can be typical if they are often accessed by a lot of people or if they offer some helpful information. Once many people have access to it, the servers are frequently overloaded, which might result in an attack. The volume and sophistication of cyber-attacks and threats are also growing as a result of this circumstance. Web-based applications are a common target for cyber-attacks since they can be accessed across a network and frequently have flaws. The challenge of web application security never ends. The moment the most recent threat is eliminated, a new one materializes. Cyber security experts find it challenging to keep track of every action taking place on the network due to the ever-increasing network density. As a result, it is necessary to automate tasks in the process of identifying web attacks. Various methods and strategies are available using machine learning to automate the detection of cyber-attacks. The methodologies to machine learning techniques for detecting attacks are discussed in this article. This paper focuses on detecting four three typical attack methods, User Datagram Protocol (UDP), Internet Control Message Protocol (ICMP), and Transmission Control Protocol-SYN (TCP-SYN) flood attacks. We evaluated Logistic Regression, K-Nearest Neighbour, MLP Classifier, and Decision Tree Classifier Machine Learning algorithms to detect web attacks in network communication flows using a continuous learning algorithm that learns the normal pattern of network traffic, and behavior of the network protocols and identify a compromised network flow. The parameters of occurrence of each assault varies, hence this study took into account three separate datasets for any and all three attack strategies.**

***Index Terms*—DDOS, securities, Machine Learning, network protocols**

I. Introduction

Internet has been the utmost important nowadays for business organizations and individuals. With the increase in demand for network-based services, network intruders have increased their attacks on these services to halt the response of services to legitimate users. Ddos Attack: The attacks which halt or slow down the services of network applications are known as Ddos attacks. A Ddos attack is achieved by attackers by controlling millions of freely available computer systems on the internet[1] .Thus causing servers to deny response to legitimate users and keeping busy with the requests generated by the attacks.

Distributed Denial of Service (DDoS) attacks have increasingly become a significant security risk that endangers the network. It uses standard protocols and services when attacking, so it is difficult to detect through traditional methods. A DDoS attack is a hostile attempt to disrupt the normal traffic of a targeted server, service, or network by overpowering the target or its neighboring infrastructure with a surge in Internet traffic. It utilizes client/server technology to merge multiple systems as an attack platform to launch attacks on one or more targets, thereby increasing the effectiveness of the attack.DDoS attacks do not follow the conventional one-to-one attack mode, making the attack behavior unique compared to other forms of attacks. It is difficult to distinguish between attack behavior and normal behavior as the attackers use standard protocols and services when attacking, resulting in difficult detection of DDoS attacks.

DDoS attack detection can be simulated as a classification problem that distinguishes between ”attack” and ”normal” network flow states[3]. After training and testing, the model predicts whether new unlabelled network traffic is benign or malicious. The classification model established, generalizes four typical attack methods: User Datagram Protocol (UDP), Internet Control Message Protocol (ICMP), Transmission Control Protocol-SYN (TCP-SYN) flood attacks and after training and testing, was able to predict whether new unlabelled network traffic is benign or malicious in real-time.DDoS attack detection techniques based on Machine Learning (ML) algorithms are becoming increasingly significant in recent times. Based on the idea of rational thinking, DDoS attack detection can be simulated as a classification problem that distinguishes between ”attack” and ”normal” network flow states. The process of processing this method includes selecting appropriate classification features to abstract the network data stream into feature vectors, assigning a label to each feature vector, and the set of labels as 0, 1. The two labels represent normal network flow and attack network flow, respectively, and choosing an appropriate classification algorithm to learn from the sample data, establish a classification model, and then use the model to classify new unlabelled network traffic as benign or malicious.The User Datagram Protocol (UDP) is a lightweight data transport protocol that works on top of IP. UDP provides a mechanism to detect corrupt data in packets, but it does not attempt to solve other problems that arise with packets, such as lost or out-of-order packets. TCP (Transmission Control

https://github.com/gakrish5/MLProject
https://github.com/rakeshrocky5430/MLProject
https://github.com/dharaniatukuri/MLProject
https://github.com/chandrakanth9933/MLProject

Protocol) is one of the main protocols of the Internet protocol suite[10]. It lies between the Application and Network Layers which are used in providing reliable delivery services. It is a connection-oriented protocol for communications that helps in the exchange of messages between the different devices over a network. ICMP is a network-level protocol. ICMP messages communicate information about network connectivity issues back to the source of the compromised transmission. It sends control messages such as destination network unreachable, source route failed, and source quench. It uses a data packet structure with an 8-byte header and variable-size data section.

The Internet Control Message Protocol (ICMP) is a network layer protocol used by network devices to diagnose network communication issues. ICMP is mainly used to determine whether or not data is reaching its intended destination in a timely manner. Commonly, the ICMP protocol is used on network devices, such as routers. ICMP is crucial for error reporting and testing, but it can also be used in distributed denial-of-service (DDoS) attacks.

User Datagram Protocol (UDP) is a Transport Layer protocol. UDP is a part of the Internet Protocol suite, referred to as UDP/IP suite. Unlike TCP, it is an unreliable and connectionless protocol. So, there is no need to establish a connection prior to data transfer. Though Transmission Control Protocol (TCP) is the dominant transport layer protocol used with most of the Internet services, it provides assured delivery, reliability, and much more but all these services cost us additional overhead and latency. Here, UDP comes into the picture. For real-time services like computer gaming, voice or video communication, and live conferences; we need UDP. Since high performance is needed, UDP permits packets to be dropped instead of processing delayed packets. There is no error checking in UDP, so it also saves bandwidth. User Datagram Protocol (UDP) is more efficient in terms of both latency and bandwidth.

To make sure that each message reaches its target location intact, the TCP/IP model breaks down the data into small bundles and afterward reassembles the bundles into the original message on the opposite end. Sending the information in little bundles of information makes it simpler to maintain efficiency as opposed to sending everything in one go.After a particular message is broken down into bundles. For example, When a user requests a web page on the internet, somewhere in the world, the server processes that request and sends back an HTML Page to that user. The server makes use of a protocol called the HTTP Protocol. The HTTP then requests the TCP layer to set the required connection and send the HTML file.Now, the TCP breaks the data into small packets and forwards it toward the Internet Protocol (IP) layer. The packets are then sent to the destination through different routes.The TCP layer in the user's system waits for the transmission to get finished and acknowledges once all packets have received.

II. Related work

Nalayini et al.[2] designed two approaches—a statistical method and a machine-learning technique—are put forth for the identification of DDoS. Entropy calculation and flow statistics analysis are implemented in the statistical technique. To calculate different thresholds, it takes into account the mean and standard deviation of the destination entropy, new flow arrival rate, packets per flow, and flow duration. The distinction between regular and attack traffic is then made using these thresholds. Random Forest classifier is used in the machine learning approach to identify DDoS attacks. They improved the Random Forest method to increase its DDoS detection accuracy.The detection and mitigation procedures are implemented using the central control and the controller's programming capabilities. It is difficult to discern between legal and malicious traffic on the Internet since usage grows daily. To increase accuracy, they specifically substitute weighted voting for traditional majority voting. The findings demonstrate that the suggested machine-learning method beats the statistical method.

Markus Ring et al. [3] offered a thorough description of the fundamental traffic and flow-based network information as well as a focused literature review of large datasets for network-based intrusion detection. There are also 15 characteristics listed that can be used to judge whether a batch of data is appropriate. These characteristics were categorised by the authors into General Information, Data Nature, Data Volume, Recording Environment, and Evaluation. They provided a thorough analysis of 35 datasets.

Rajesh et al. [1] suggested a solution that employs machine learning methods to detect malware and ddos attacks with high detection accuracy. A DDOS attack is defined by the blockage of a certain service by rendering the victim's resources useless for their intended use, which causes server failure. DDoS generates attacks by turning networked devices into remotely controlled bots.A private cloud's virtual instances are used to create the real-time traffic. By taking into account the numerous snmp parameters and categorising using machine learning techniques like bagging, boosting, and ensemble models, the ddos attack is recognised. Additionally, different malware on networked devices is prevented from being exploited as a bot to generate ddos attacks.

Gaurav Somani et al. [6] produced a thorough and in-depth analysis of DDoS attacks and potential defence strategies for the cloud computing environment. Through the conversation, they have demonstrated that the most common type of DDoS attack in the cloud is the EDoS attack. DDoS attacks have significant traits that are crucial when evaluating utility computing models. In their study, they discussed cloud computing elements that are essential for comprehending DDoS attacks and their effects.

Peng Xia et al.[14] proposed to analyze the data centre flow correlation information first. Second, they offered a reliable detection method for DDoS attacks based on CKNN (K-nearest neighbors traffic categorization with correlation analysis). The data centre is greatly threatened by distributed denial-of-service (DDoS) assaults, and numerous protection measures have been proposed to identify them. In an effort to reduce the enormous cost, they also introduced the r-polling method, a grid-based approach that uses less training data. The methodology is effective at identifying aberrant traffic with high efficiency, cheap cost, and a broad detection range when compared to previous methods.

Zekri et al.[10] created a DDoS detection system based on the C. 4.5 algorithm . This algorithm creates a decision tree to automatically and effectively detect signature attacks for DDoS flooding attacks when used in conjunction with signature detection techniques. DDoS (Distributed Denial of Service) attacks are among the most prevalent ones that cause significant harm and impair cloud functionality. In a DDoS attack, the attacker typically employs innocent computers that have been compromised (referred to as zombies) in order to transmit a high number of packets from these already-captured zombies to a server by taking advantage of known or undisclosed faults and vulnerabilities. This could take up a significant amount of the victim cloud infrastructures' network bandwidth or take up a lot of the servers' time.

Cheng ,et al.[7] created an IP address database using a sequential storage architecture with constant time complexity. If and only if the quantity of continuous PDRA sequence values that all surpass a PDRA abnormal threshold (PAT) crosses a predetermined threshold, the autoregressive integrated moving average (ARIMA) trending prediction module will be activated. The likelihood of the forecasted PDRA sequence value exceeding the PAT should then be calculated. Finally, using the aberrant likelihood of the forecasted PDRA sequence, they detect the DDoS attack. The method they suggested can effectively reduce the use of computational resources, identify DDoS assault at its earlier stage with a greater detection rate, and reduce the number of false alarms, according to both theorem and experiment.

Idhammad et al.[9] presented network entropy estimation, co-clustering, information gain ratio, and extra-trees algorithm-based online sequential semi-supervised ML technique for DDoS detection.Despite the use of cutting-edge Machine Learning (ML) algorithms for DDoS detection, the attack still poses a serious threat to the Internet. Most ML-based DDoS detection methods now in use fall into one of two categories: supervised or unsupervised. Supervised ML methods for DDoS detection rely on labelled network traffic datasets being available. Unsupervised ML methods, on the other hand, identify attacks by examining incoming network traffic. The approach's unsupervised component enables the reduction of irrelevant regular traffic data for DDoS detection, improving accuracy and decreasing the number of false positives. While the supervised part enables more accurate classification of DDoS traffic and a decrease in the unsupervised half's false positive rates.

III. Motivation

Web application security is a never-ending game of cat and mouse. A web application is a computer program that utilizes web browsers and web technology to perform tasks over the Internet. Basically, a web application requires a web server to manage requests from the client, an application server to perform the tasks requested, and a database to store the information. Web attacks can be a serious threat to a web application. They take advantage of the vulnerability of the application to gain access to the application DB and do serious damage. The volume and sophistication of cyber-attacks and threats are also growing as a result of this circumstance. Web-based applications are a common target for cyber-attacks since they can be accessed across a network and frequently have flaws. The challenge of web application security never ends. The moment the most recent threat is eliminated, a new one materializes. Cybersecurity experts find it challenging to keep track of every action taking place on the network due to the ever-increasing network density. This motivated us to automate the process of Web Attacks Detection using ML. ML Algorithms are capable of learning a large amount of malicious and benign requests of different patterns and can predict them effectively in production.

IV. Objectives

This paper is focused on evaluating different Machine Learning Algorithms for the detection of Ddos-Attack of following types:

- User Datagram Protocol (UDP) flood Attacks.
- Transmission Control Protocol-SYN flood attacks.
- Internet Control Message Protocol (ICMP) flood Attacks.

For this, we have evaluated the following algorithms for each attack type:

- Logistic Regression.
- MLP CLassifier.
- KNN Classifier.
- Decision Tree Classifier.

V. Proposed frame work

The SDN dataset is considered and as a first step, the data is preprocessed. After data preprocessing, feature extraction is done. This data is trained to different machine learning models such as Logistic Regression, K-Nearest Neighbour, MLP Classifier, and Decision Tree Classifier. After training, testing is done. Finally the result analysis is done. The whole methodology is depicted in Fig. 1

A. Dataset Collection

Here we have collected three different datasets for attacks on three different protocols. The dataset collected for ICMP protocol[15] has a size of 311029 X 42. This dataset consists of parameters such as duration, service, src-bytes, wrong-fragment, count, urgent, hot...etc.The dataset collected for TCP protocol[16] has a size of 311029 X 42. This dataset consists of parameters such as duration, protocol_type, service, flag, src_bytes, dst_bytes, land, wrong_fragment, urgent..etc.Here protocol_type is fixed to TCP. The dataset collected for UDP protocol[17] has a size of 26703 X 42. This dataset consists of parameters such as duration, protocol_type, service flag, src_bytes, dst_bytes, land, wrong_fragment, urgent..etc.Here protocol_type is fixed to UDP.

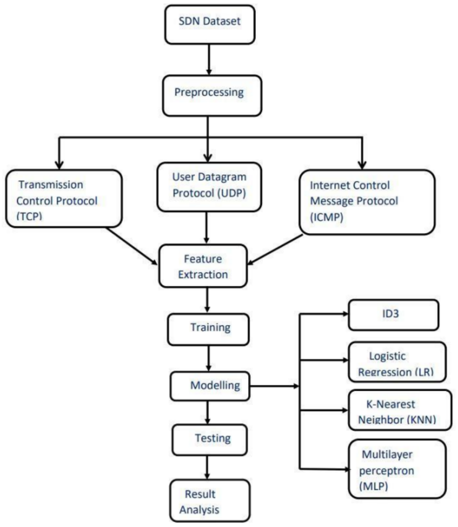


Fig. 1. Process Flow Diagram

B. Model building

Here in this section we used sklearn module from python to impelment the machine learning algorithms.

1) Logistic regression: It is a statistical method for analyzing a data set in which there are one or more independent variables that determine an outcome. The outcome is measured with a dichotomous variable (in which there are only two possible outcomes). The goal of logistic regression is to find the best fitting model to describe the relationship between the dichotomous characteristic of interest (dependent variable = response or outcome variable) and a set of independent (predictor or explanatory) variables.

The "equation(1)" for logistic regression is as follows:

$$ProbabilityOfOutcome(\hat{Y}_i) = \frac{e^{\beta_0+\beta_1X_1+\beta_2X_2+...+\beta_iX_i}}{1+e^{\beta_0+\beta_1X_1+\beta_2X_2+...+\beta_iX_i}} \quad (1)$$

- Here $\hat{Y}$i represents the estimated probability of being in one binary outcome category (i) versus the other
- $e^{\beta_0+\beta_1X_1+\beta_2X_2+...+\beta_iX_i}$ represents the linear regression equation for independent variables expressed in the logistic scale

The logistic scale transforms the original equation of linear regression to obtain natural log of the odds of being in one outcome category ($\hat{Y}$) versus the other category (1 –$\hat{Y}$) is given by "(2)"

$$ln(\hat{Y}/1–\hat{Y}) = \beta_0 + \beta_1X_1 + \beta_2X_2 + \ldots + \beta_iX_i \quad (2)$$

Despite its name, logistic regression is more of a classification model than a regression model[18]. For situations involving binary and linear classification, logistic regression is a straightforward and more effective approach. It's a classification model that's incredibly simple to implement and performs admirably with linearly separable classes.

2) K-Nearest Neighbour: The k-nearest-neighbors algorithm is a classification algorithm, and it is supervised: it takes a bunch of labelled points and uses them to learn how to label other points. To label a new point, it looks at the labelled points closest to that new point (those are its nearest neighbors), and has those neighbors vote, so whichever label the most of the neighbors have is the label for the new point (the "k" is the number of neighbors it checks). The k-nearest neighbors (KNN) algorithm is a simple[19], supervised machine learning algorithm that can be used to solve both classification and regression problems. It's easy to implement and understand, but has a major drawback of becoming significantly slows as the size of that data in use grows.

3) MLP Classifier: Multi layer perceptron (MLP) is a supplement of feed forward neural network. It consists of three types of layers: the input layer, output layer and hidden layer, The input layer receives the input signal to be processed. The required task such as prediction and classification is performed by the output layer. An arbitrary number of hidden layers that are placed in between the input and output layer are the true computational engine of the MLP. Similar to a feed forward network in a MLP the data flows in the forward direction from input to output layer. The neurons in the MLP are trained with the back propagation learning algorithm. MLPs are designed to approximate any continuous function and can solve problems which are not linearly separable. The major use cases of MLP are pattern classification, recognition, prediction and approximation.

4) Decision Tree: Decision tree builds classification or regres- sion models in the form of a tree structure. It breaks down a data set into smaller and smaller subsets while at the same time an associated decision tree is incrementally developed. The final result is a tree with decision nodes and leaf nodes. A decision node has two or more branches and a leaf node represents a classification or decision. The topmost decision node in a tree which corresponds to the best predictor called root node. Decision trees can handle both categorical and numerical data

VI. Results and Analysis

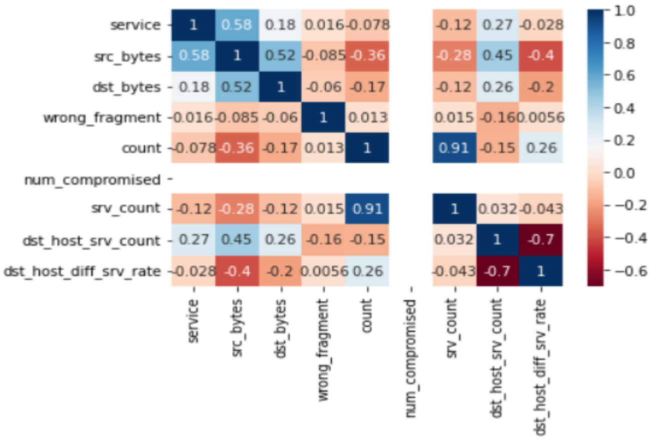
The main goal of this article is to detect cyber attacks that occur in three different ways: User Datagram Protocol flood attacks, Internet Control Message Protocol flood attacks, and Transmission Control Protocol-SYN flood attacks. For detection of each of these attacks, we evaluated four continuous machine learning algorithms, they are Logistic Regression, MLP classifier, KNN, and Decision Tree Classifier.

A. Predicting The User Datagram Protocol (UDP) flood attacks.

To detect the UDP flood attacks, The following four models have been employed. Keeping the precision and recall of the model in mind, instead of dumping all the attributes in the dataset into the model, the models are trained on specific attributes which are described below.

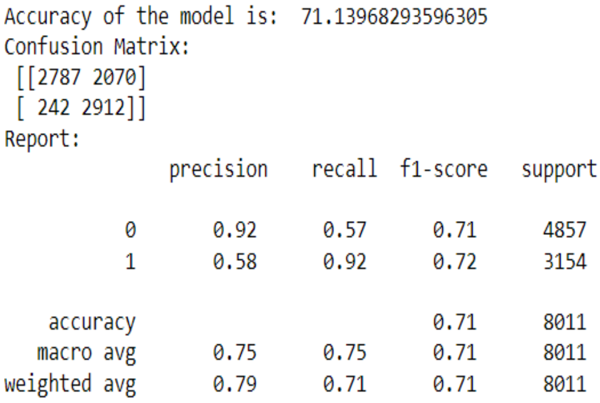
Predictor Variables: service, src_bytes, dst_bytes,

wrong_fragment, count, num_compromised, srv_count, dst_host_srv_count, dst_host_diff_srv_rate.
Response Variable: result.



	service	src_bytes	dst_bytes	wrong_fragment	count	num_compromised	srv_count	dst_host_srv_count	dst_host_diff_srv_rate
service	1	0.58	0.18	0.016	-0.078		-0.12	0.27	-0.028
src_bytes	0.58	1	0.52	-0.085	-0.36		-0.28	0.45	-0.4
dst_bytes	0.18	0.52	1	-0.06	-0.17		-0.12	0.26	-0.2
wrong_fragment	-0.016	-0.085	-0.06	1	0.013		0.015	-0.16	0.0056
count	-0.078	-0.36	-0.17	0.013	1		0.91	-0.15	0.26
num_compromised									
srv_count	-0.12	-0.28	-0.12	0.015	0.91		1	0.032	-0.043
dst_host_srv_count	0.27	0.45	0.26	-0.16	-0.15		0.032	1	-0.7
dst_host_diff_srv_rate	-0.028	-0.4	-0.2	0.0056	0.26		-0.043	-0.7	1

Fig. 2. Correlation matrix for UDP attack dataset features

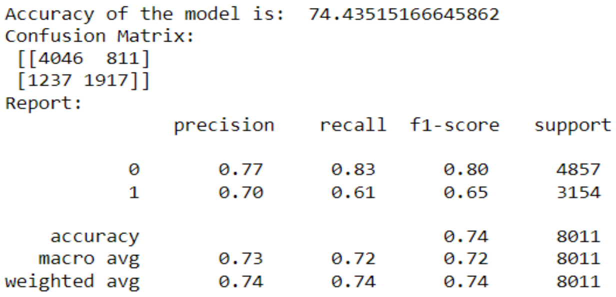


```
Accuracy of the model is:  71.13968293596305
Confusion Matrix:
 [[2787 2070]
 [ 242 2912]]
Report:
               precision    recall  f1-score   support

           0       0.92      0.57      0.71      4857
           1       0.58      0.92      0.72      3154

    accuracy                           0.71      8011
   macro avg       0.75      0.75      0.71      8011
weighted avg       0.79      0.71      0.71      8011
```

Fig. 3. Metrics obtained when Logistic Regression is applied



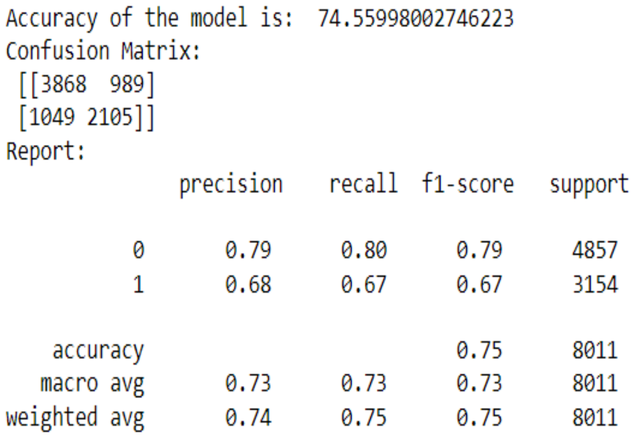
```
Accuracy of the model is:  74.43515166645862
Confusion Matrix:
 [[4046  811]
 [1237 1917]]
Report:
               precision    recall  f1-score   support

           0       0.77      0.83      0.80      4857
           1       0.70      0.61      0.65      3154

    accuracy                           0.74      8011
   macro avg       0.73      0.72      0.72      8011
weighted avg       0.74      0.74      0.74      8011
```

Fig. 4. Metrics obtained when KNN is applied

The comparison of all the above models has been graphically represented in Fig. 7

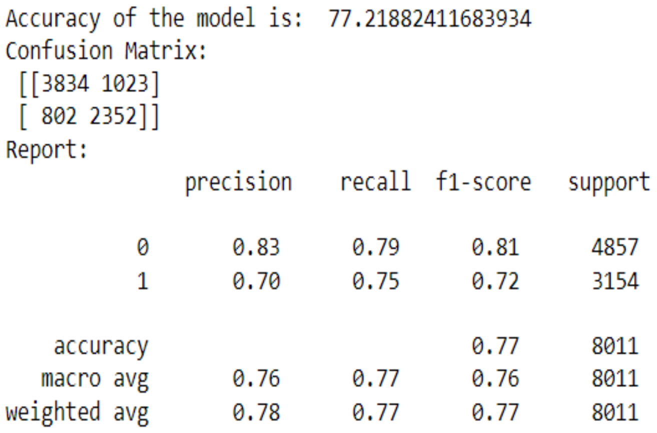


```
Accuracy of the model is:  74.55998002746223
Confusion Matrix:
 [[3868  989]
 [1049 2105]]
Report:
               precision    recall  f1-score   support

           0       0.79      0.80      0.79      4857
           1       0.68      0.67      0.67      3154

    accuracy                           0.75      8011
   macro avg       0.73      0.73      0.73      8011
weighted avg       0.74      0.75      0.75      8011
```

Fig. 5. Metrics obtained when MLP is applied



```
Accuracy of the model is:  77.21882411683934
Confusion Matrix:
 [[3834 1023]
 [ 802 2352]]
Report:
               precision    recall  f1-score   support

           0       0.83      0.79      0.81      4857
           1       0.70      0.75      0.72      3154

    accuracy                           0.77      8011
   macro avg       0.76      0.77      0.76      8011
weighted avg       0.78      0.77      0.77      8011
```

Fig. 6. Metrics obtained when ID3 is applied

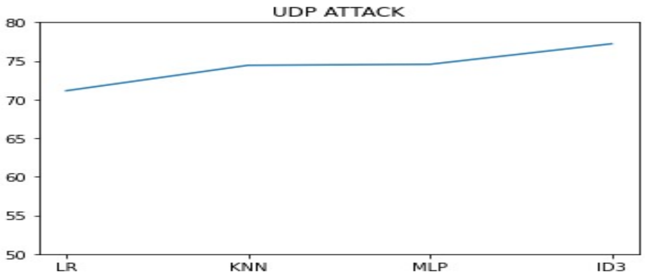


Fig. 7. Metrics Comparision of LR, KNN, MLP, and ID3 for UDP Attack

B. Predicting The Transmission Control Protocol-SYN flood attacks.

To detect the TCP-SYN flood attacks, The following four models have been employed. Keeping the precision and recall of the model in mind, instead of dumping all the attributes in the dataset into the model, the models are trained on specific attributes which are described below. Predictor Variables: service, count", srv_count, src_bytes, serror_rate. Response Variable: result.

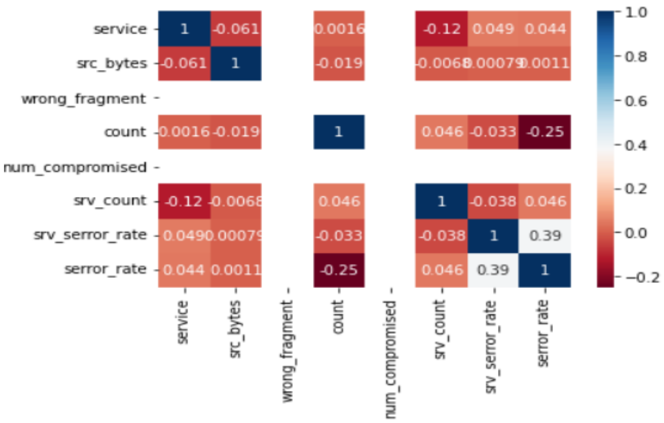
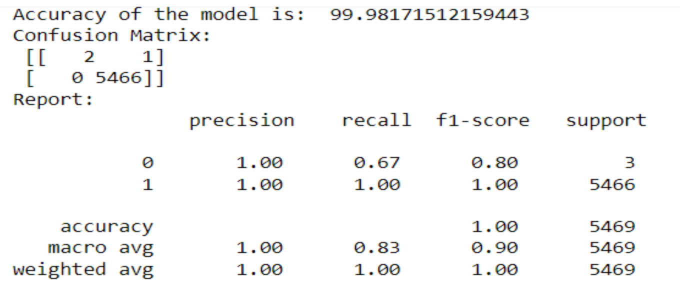


Fig. 8. Correlation matrix for TCP-SYN attack dataset features

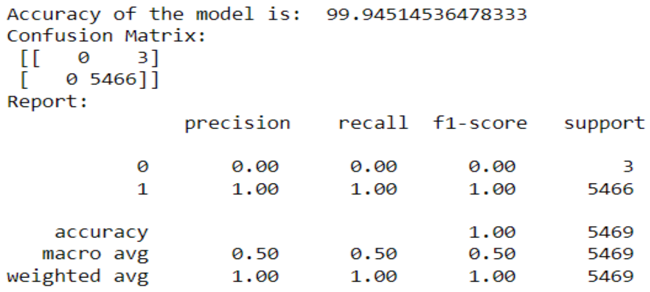


```
Accuracy of the model is:   99.98171512159443
Confusion Matrix:
 [[    2     1]
 [    0 5466]]
Report:
               precision    recall  f1-score   support

           0       1.00      0.67      0.80         3
           1       1.00      1.00      1.00      5466

    accuracy                           1.00      5469
   macro avg       1.00      0.83      0.90      5469
weighted avg       1.00      1.00      1.00      5469
```

Fig. 9. Metrics obtained when KNN is applied



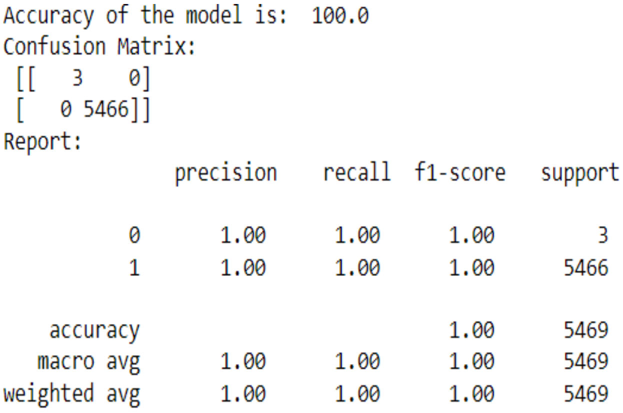
```
Accuracy of the model is:   99.94514536478333
Confusion Matrix:
 [[    0     3]
 [    0 5466]]
Report:
               precision    recall  f1-score   support

           0       0.00      0.00      0.00         3
           1       1.00      1.00      1.00      5466

    accuracy                           1.00      5469
   macro avg       0.50      0.50      0.50      5469
weighted avg       1.00      1.00      1.00      5469
```

Fig. 10. Metrics obtained when MLP is applied

The performance of MLP, KNN and ID3 on TCP-SYN dataset is graphically described in Fig. 12



```
Accuracy of the model is:  100.0
Confusion Matrix:
 [[    3    0]
 [    0 5466]]
Report:
               precision    recall  f1-score   support

           0       1.00      1.00      1.00         3
           1       1.00      1.00      1.00      5466

    accuracy                           1.00      5469
   macro avg       1.00      1.00      1.00      5469
weighted avg       1.00      1.00      1.00      5469
```

Fig. 11. Metrics obtained when ID3 is applied

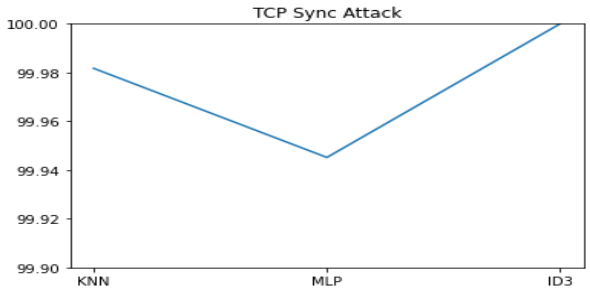


Fig. 12. Metrics Comparision of KNN, MLP, and ID3 for TCP-SYN Attack

C. Predicting The Internet Control Message Protocol flood attacks.

To detect the ICMP flood attacks, The following four models have been employed. Keeping the precision and recall of the model in mind, instead of dumping all the attributes in the dataset into the model, the models are trained on specific attributes which are described below.
Predictor Variables: duration, service, src_bytes, wrong_fragment, count, urgent, num_compromised, srv_count.
Response Variable: result.

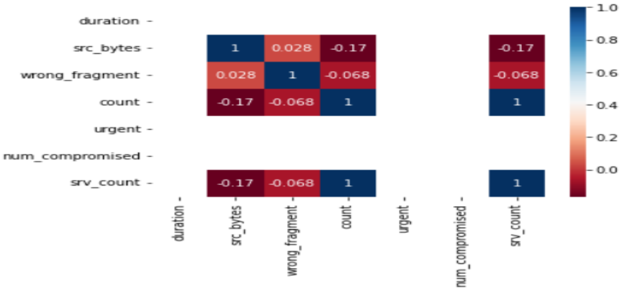
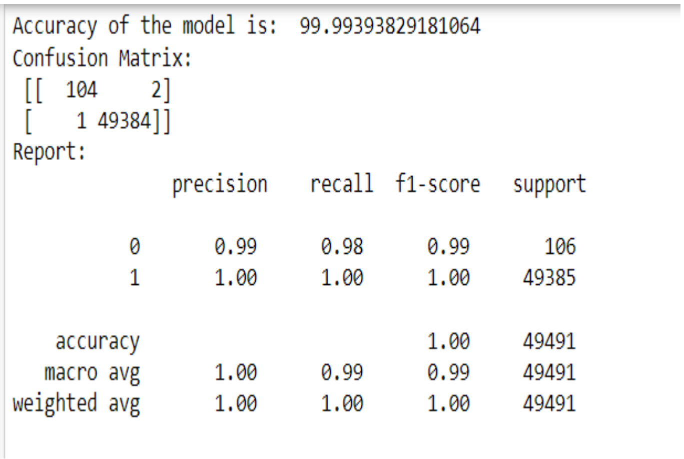


Fig. 13. Correlation matrix for ICMP attack dataset features

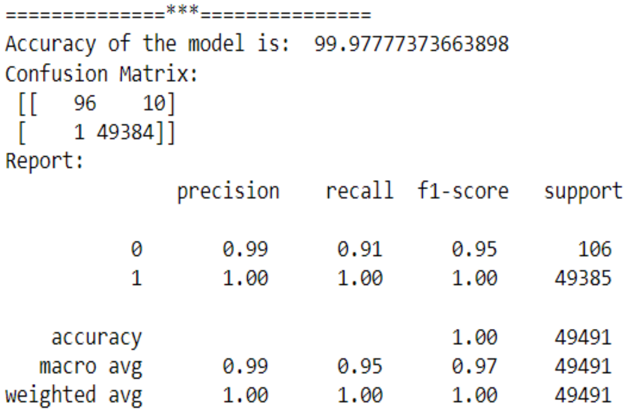


```
Accuracy of the model is:  99.99393829181064
Confusion Matrix:
 [[  104     2]
 [    1 49384]]
Report:
              precision    recall  f1-score   support

           0       0.99      0.98      0.99       106
           1       1.00      1.00      1.00     49385

    accuracy                           1.00     49491
   macro avg       1.00      0.99      0.99     49491
weighted avg       1.00      1.00      1.00     49491
```

Fig. 14. Metrics obtained when KNN is applied

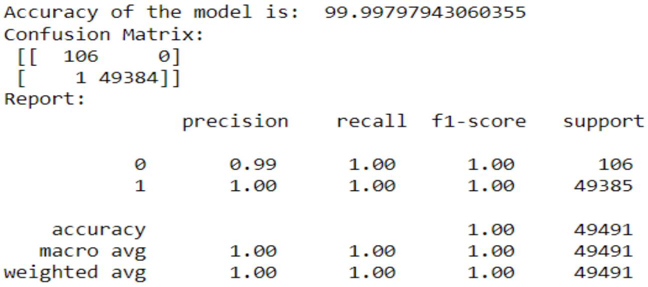


```
==============***===============
Accuracy of the model is:  99.97777373663898
Confusion Matrix:
 [[   96    10]
 [    1 49384]]
Report:
              precision    recall  f1-score   support

           0       0.99      0.91      0.95       106
           1       1.00      1.00      1.00     49385

    accuracy                           1.00     49491
   macro avg       0.99      0.95      0.97     49491
weighted avg       1.00      1.00      1.00     49491
```

Fig. 15. Metrics obtained when MLP is applied



```
Accuracy of the model is:  99.99797943060355
Confusion Matrix:
 [[  106     0]
 [    1 49384]]
Report:
              precision    recall  f1-score   support

           0       0.99      1.00      1.00       106
           1       1.00      1.00      1.00     49385

    accuracy                           1.00     49491
   macro avg       1.00      1.00      1.00     49491
weighted avg       1.00      1.00      1.00     49491
```

Fig. 16. Metrics obtained when ID3 is applied

The performance of MLP, KNN and ID3 on ICMP dataset is graphically described in Fig. 17

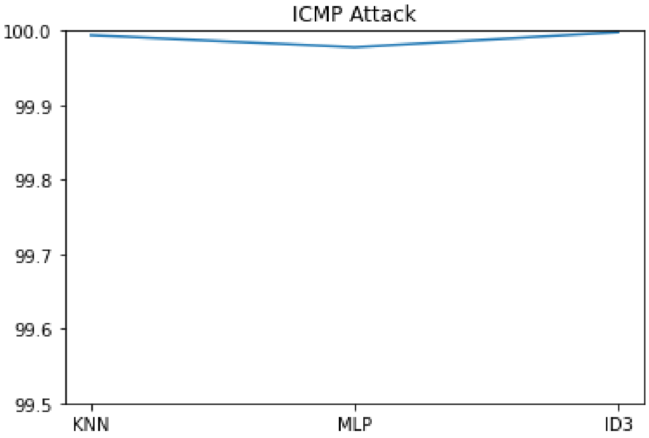
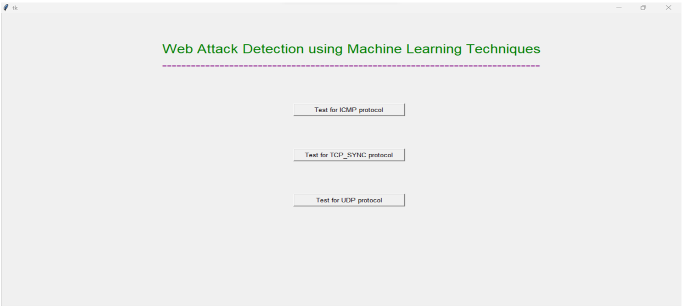


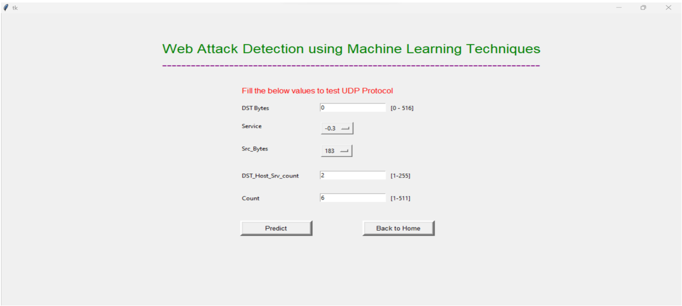
Fig. 17. Metrics Comparision of KNN, MLP, and ID3 for ICMP Attack

D. Interface



Web Attack Detection using Machine Learning Techniques

Fig. 18. Home Page in the Interface



Web Attack Detection using Machine Learning Techniques

Fill the below values to test UDP Protocol

DST Bytes 0 [0 - 516]

Service -0.3

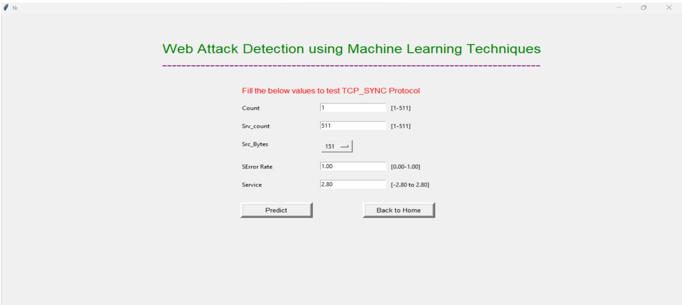
Src_Bytes 183

DST_Host_Srv_count 2 [1-255]

Count 6 [1-511]

Fig. 19. UDP Attack Prediction page

We developed an interface using Tkinter and binded the best model for each of the attack types in the backend. Here, the user needs to select the type of attack that he/she would like to detect. Fig. 19, Fig. 20, Fig. 21 displays the UDP, TCP-SYN, and ICMP attacks prediction pages consisting of the parameters on which the model got trained. All pages in the Interface consist of two buttons, the Predict Button, clicking which the user can know whether there is any Attack Not and

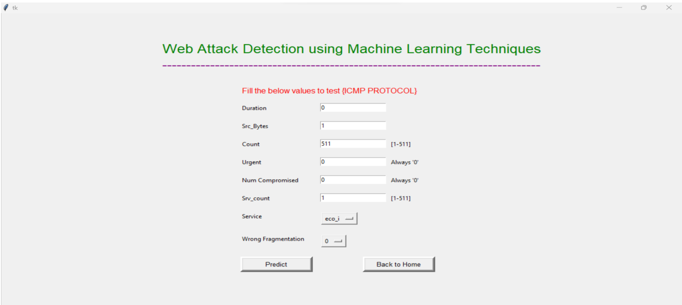


Web Attack Detection using Machine Learning Techniques

Fill the below values to test TCP_SYNC Protocol

Count 1 [1-511]
Srv_count 511 [1-511]
Src_Bytes 151
SError Rate 1.00 [0.00-1.00]
Service 2.80 [-2.80 to 2.80]

Fig. 20. TCP-SYN Attack Prediction page



Web Attack Detection using Machine Learning Techniques

Fill the below values to test {ICMP PROTOCOL}

Duration 0
Src_Bytes 1
Count 511 [1-511]
Urgent 0 Always '0'
Num Compromised 0 Always '0'
Srv_count 1 [1-511]
Service eco_i
Wrong Fragmentation 0

Fig. 21. ICMP Attack Prediction page

the Back TO Home Page button, which upon clicking redirects the user to page in Fig. 18

Conclusion

This paper evaluated Logistic Regression, K-Nearest Neighbor, MLP, and Decision Tree Classifiers to detect the possibility of UDP, ICMP, and TCP-SYN flood attacks. We also developed an interface using Tkinter by binding the best model for each of the attack types with the interface. We presented a detailed performance comparison of all the considered models for each considered attack type.

References

[1] Rajesh, S., Clement, M., SB, S., SH, A. S., Johnson, J. (2021). Real-Time DDoS Attack Detection Based on Machine Learning Algorithms. Available at SSRN 3974241.

[2] M NALAYINI, C., Katiravan, J. (2022). Detection of DDoS Attack Using Machine Learning Algorithms. Available at SSRN 4173187, 9(7).

[3] Ring, M., Wunderlich, S., Scheuring, D., Landes, D., Hotho, A. (2019). A survey of network-based intrusion detection data sets. Computers Security, 86, 147-167.

[4] D. Chaudhary, Bhushan, K., Gupta, B. B. (2018). Survey on DDoS attacks and defense mechanisms in cloud and fog computing. International Journal of E-Services and Mobile Applications (IJESMA), 10(3), 61-83.

[5] Salim, M. M., Rathore, S., Park, J. H. (2020). Distributed denial of service attacks and its defenses in IoT: a survey. The Journal of Supercomputing, 76(7), 5320-5363.

[6] Somani, G., Gaur, M. S., Sanghi, D., Conti, M., Buyya, R. (2017). DDoS attacks in cloud computing: Issues, taxonomy, and future directions. Computer Communications, 107, 30-48.

[7] Cheng, J., Xu, R., Tang, X., Sheng, V. S., Cai, C. (2018). An abnormal network flow feature sequence prediction approach for DDoS attacks detection in big data environment. Comput. Mater. Continua, 55(1), 95-119.

[8] Yuan, X., Li, C., Li, X. (2017, May). DeepDefense: identifying DDoS attack via deep learning. In 2017 IEEE international conference on smart computing (SMARTCOMP) (pp. 1-8). IEEE.

[9] Idhammad, M., Afdel, K., Belouch, M. (2018). Semi-supervised machine learning approach for DDoS detection. Applied Intelligence, 48(10), 3193-3208.

[10] Zekri, M., El Kafhali, S., Aboutabit, N., Saadi, Y. (2017, October). DDoS attack detection using machine learning techniques in cloud computing environments. In 2017 3rd international conference of cloud computing technologies and applications (CloudTech) (pp. 1-7). IEEE.

[11] Wang, T. S., Lin, H. T., Cheng, W. T., Chen, C. Y. (2017). DBod: Clustering and detecting DGA-based botnets using DNS traffic analysis. Computers Security, 64, 1-15.

[12] Lakshminarayanan, K., Adkins, D., Perrig, A., Stoica, I. (2004). Taming IP packet flooding attacks. ACM SIGCOMM Computer Communication Review, 34(1), 45-50.

[13] Gligor, V. D. (1984). A note on denial-of-service in operating systems. IEEE Transactions on Software Engineering, (3), 320-324.

[14] Xiao, P., Qu, W., Qi, H., Li, Z. (2015). Detecting DDoS attacks against data center with correlation analysis. Computer Communications, 67, 66-74.

[15] Stoltzfus, J. C. (2011). Logistic regression: a brief primer. Academic emergency medicine, 18(10), 1099-1104.

[16] Guo, G., Wang, H., Bell, D., Bi, Y., Greer, K. (2003, November). KNN model-based approach in classification. In OTM Confederated International Conferences" On the Move to Meaningful Internet Systems" (pp. 986-996). Springer, Berlin, Heidelberg.

[17] R. Rajaram, V. U. Shastry, and R. U. Acharya, "DDoS attack detection using supervised machine learning techniques," in Proceedings of the 2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI), Udupi, India, 2017, pp. 2045-2049.

[18] S. S. Mohapatra, S. K. Mishra, and S. K. Rath, "Machine Learning based Detection and Mitigation of DDoS Attacks: A Survey," Journal of Information Security, vol. 9, no. 1, pp. 1-13, 2018.

[19] H. Shah, R. Ranjan, A. Swami, and S. K. Jaiswal, "A Review of Machine Learning Techniques for DDoS Attack Detection," in Proceedings of the 2019 11th International Conference on Computational Intelligence and Communication Networks (CICN), Dhanbad, India, 2019, pp. 1-6.

[20] Qianmu Li, Yanjun Song, Jing Zhang, Victor S. Sheng. "Multiclass imbalanced learning with one versus-one decomposition and spectral clustering". Expert Systems with Applications, Volume 147,2020.

[21] Kleinbaum, D. G., Dietz, K., Gail, M., Klein, M., Klein, M. (2002). Logistic regression (p. 536). New York: Springer-Verlag.