Detection of DDoS attacks using Machine Learning

Ms. J Anusha[1], Mr.B. Akhil Krishna[2] ,U. Mr. Munindhar[3], Ms.K Vaishnavi[4], Ms.B.Srinidhi[5]
(1) Assistant Professor of ECE Dept. & (2,3,4,5) Students,
Department of Electronics & Communication Engineering, KITS(S), Huzurabad, Karimangar, T.S..
Mail ID: j.anushaece@gmail.com

ABSTRACT

Distributed Denial of Service (DDoS) assaults are cyberattacks that use numerous computers to transmit massive data packets that exhaust the resources of the computer network services. The computer network service's port mirroring allows for the observation and capture of the entire data packet as well as significant data are in log files format delivered by attacker. Network traffic divided into two conditions: regular traffic and attack traffic, according to the classification system.

Various machine learning methods, including support vector machines have been used in this research. The random forest model outperformed traditional algorithms in terms of performance. We used the Canadian Institute for Cyber Security (CIC) dataset to train these algorithms. It covers 10 possible attacks of IOT environment and normal class. One approach for processing numerical attributes as input and determining whether access to a network will be "normal" or "attack" access by DDoS, is the Random Forest classification.

The purpose of the research is to train a model using machine learning approaches that can detect and categorize the type of DDoS assault with more accuracy than each individual machine learning technique utilized.

Keywords: DDoS Attack, IOT Environment, SDN, Forest Fire, LoRa.

Introduction

The new SDN architecture environment offers deep packet inspection via the whole network perspective. It promotes quick action and regular updates to traffic laws and regulations. The SDN is capable of service-open intelligent scheduling, flexible and schedule-able rapid deployment, and perceived control over the total visualization perspective. By network service and reducing physical measurement losses cost, the software defined network enhances user experience and makes it simpler to promote the implementation of the complete network. Researchers who concentrated on conventional network design presented a number of DDoS assault detection methods. Software-defined networking (SDN) has created a revolutionary architecture that separates control and dataplanes, which are generally combined in conventional network. The data layer, the control layer, and the application layer are the three main layers into which SDN divides a network. While being cognitively incompetent in the data plane, SDN switches are managed by a centralized controller in the control plane. The ease of management makes the advantages of this revolution obvious. However, there is a big chance that the controller could end up being the only point of failure. In other words, if the controller goes down, all the SDN switches that are connected to it may lose functionality if they can no longer communicate with it. The control and data planes are connected through the Open Flow protocol, which also consults the controller on how to handle particular packets.

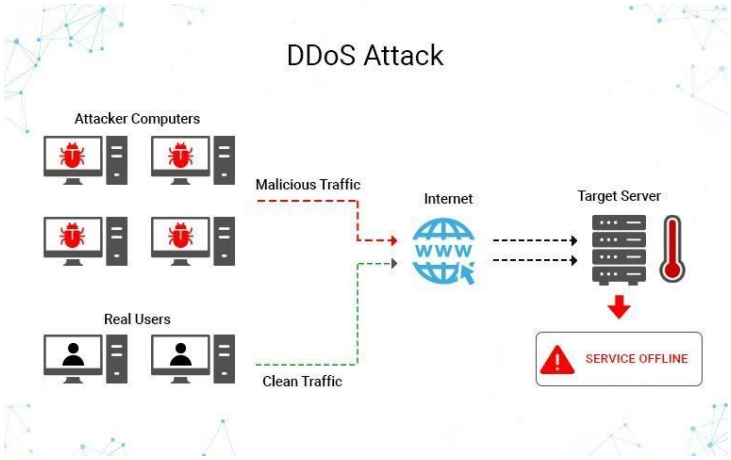


Fig: 1 DDos Attack

Data sharing and machine-to-machine connectivity are made feasible by IoT, which broadens the coverage. IoT intends to combine a variety of hardware and networks to make localization, monitoring, management, and other functions possible through sensor identification and pervasive computing. The enormous demand for data collection and environmental monitoring has caused an exponential rise in the number of devices connected to the network in outdoor deployments. New technologies have recently surfaced that promise to provide low-power and long-range connectivity options for Internet of Things applications, including LoRa, DASH7, and Narrowband (NB-IoT). The key requirements of price, battery life, coverage area, scalability, security, and privacy will also be met by these technologies, according to their promises. Because there are so many IoT devices connected to one network, expanding the attack surface, IoT networks now confront new security challenges.

Problem Overview

The services of networks with essential business and industry information have spread to the production and life of contemporary society as an outcome of the continuous advancement of network technology, the endless expansion of network business requirements, and the explosive growth of the digital economy in the Internet age. When DDoS attacks start to appear, the corresponding network services may experience irregularities. This could have disastrous repercussions, including significant financial losses. DDoS assaults are one of the main threats to network security that the Internet faces. In the security sector, accurate and quick DDoS attack detection is a key research subject. The network data plane and the control plane, which offers network programmability and centralized administration control, are separated by the new network innovation architecture known as SDN. Attackers on the network target application resources, system resources, and network bandwidth to execute denial of service assaults.

The difficulties in identifying DDoS assaults include the following:

- Difficult to determine attack traffic characteristics;
- Insufficient cooperation between coherence network nodes
- A lower barrier to usage strengthens the assault tool.

- Widespread address fraud makes it challenging to pinpoint the assault's origin, and attack length and reaction time restrictions.

The two primary DDoS attack detection techniques used in conventional network design are attack detection based on traffic characteristics and attack detection based on anomalous traffic patterns. By compiling various sorts of attack characteristic data, the former creates a database of DDoS assault characteristics. It can determine whether a network is being attacked by DDoS by comparing and analyzing the data from the most recent network data packet and characteristics database. Characteristics matching, model reasoning, state transition, and expert systems are the primary implementation techniques. The latter's primary objective is to create a traffic model and look into unusual flow variations in order to determine whether the traffic is abnormal or not and whether the server has been attacked.

Literature Survey

Lin and Wang [5] developed an SDN-based DDoS attack detection and defense mechanism, however the deployment and operation were challenging owing to the method's usage of three Open Flow management tools that used the sFlow standard for anomaly detection. Yang et al.'s [6] approach for identifying IP traffic with a better and more precise detection effect integrates both the flow information and the IP entropy characteristic information using a single flow information and information. Despite the adaptability and practicality of information entropy, other technologies are still required to calculate the threshold and multi element weight distribution.

By analyzing the characteristics of each TCP/UDP/ICMP protocol through the training ANN algorithm, Saied et al. [7] proposed the notion that the technique needs to identify packet protocol, which is complicated and wasteful, to detect DDoS assaults. The SOM approach is used to recognize DDoS assaults by locating the flow data connected to them. This method uses less energy while offering a high detection rate. The most important element is the time period's extraction. The disadvantage of this method is that there is some hysteresis in the detection and the assault behavior is not quickly and accurately detected.

It is proposed a framework for detecting and mitigating DDoS attacks in a large-scale network, however it is not appropriate for small-scale implementation.

It is advised to utilise a DDoS attack detection system that includes a database of true source and destination IP addresses. Based on the nonparametric cumulative method CUSUM, the strategy must be adjusted and the threshold selected. When a DDoS assault happens, it analyses the unusual qualities of the source IP address and the destination IP address and rapidly validates the DDoS attack.

The SOM algorithm must predict the amount of neurons due to the high information entropy false positive rate. In order to classify each attack, we first identify the traits of various DDoS attacks, gather information from switch flow tables, extract the six-tuple characteristic values matrix, and then develop an SVM classification model. The method may analyze multidimensional data and transform low-dimensional data that isn't linearly separable into a high-dimensional feature space so that it can be linearly separable and reliably classified. The approach is currently widely used in anomaly detection and classification. Dao et al. [12] build a table in the controller to track packets by IP address during a DDoS attack. The number of packets using that connection is also compared with a minimum value in order to distinguish between a valid request and an attack. The simulation shows that this approach successfully reduces flow entries in the

switch while maintaining the bandwidth of the controller-switch channel under DDoS attacks. This strategy takes a lot of controller resources if the attacker changes the source address. Mousavi and St-Hilaire [13] suggest utilizing entropy to measure unpredictability, where time period and threshold are two important factors, in order to detect DDoS. While countermeasures might improve detection accuracy in the actual network, the solutions that are being offered only deal with detection. Dong et al. advise using the Sequential Probability Ratio Test (SPRT), a statistical technique. [14] to address the false positive and false negative issues that are currently present. The decision is made using the log-probability ratio and predefines two boundaries (A and B, B A) relating to the probabilities of false positive (a) and false negative (b); it is suggested that A = b/(1 a), and B = (1 b)/(1 a). The DARPA Intrusion Detection Data Sets analysis shows how rapid and precise it is. However, the proposed method is evaluated exclusively on the basis of mathematical results rather than simulations, which can contain random variables.

Yan et al. [15] propose the "Multislot" method to execute requests in each time slot so that legitimate users can correctly communicate with one another during DDoS attacks.

Low rate DDOS attack detection:

Low-rate Distributed Denial-of-Service provide a threat to the internet since they hinder legal traffic by delivering numerous attack packets that are identical to other types of traffic, hence causing congestion. Zhang and associates. It was suggested to use a congestion-participation (CPR) metric and CPR-based technique to detect and filter DDoS attacks at low rates. They found that while low-rate DDoS attacks actively contribute to network congestion, ordinary TCP flows actively relieve it. The suggested method was created to distinguish between attack flows and legitimate flows. To assess its effectiveness, more study and analysis with real datasets is required. This was done on the assumption that the typical packet sizes of many applications rely on data requests, data answers, and data acknowledgments. The recommended approach can only be scaled up to a certain degree owing to its dependency on the packets in the observation window and the requirement for a lengthy detection time to obtain a high likelihood of detection despite being exposed to a low-rate DDoS assault. Entropy-based DDoS assault low-level detection was proposed by Jadhav and Patil[5] as an effective, impartial technique. Compared to using traditional metric entropy, this method performs significantly better. The false positive rate is increasing, while there is a very slight distinction between regular traffic and attack traffic.

High rate DDOS attack detection:

The defenses against spoof DDoS attacks have undergone a thorough investigation. Each tactic has advantages and disadvantages. To launch a phony DDoS assault, the attacker alters TCP/IP header information. The Time-To-Live (TTL) field cannot be faked, unlike other TCP/IP header information. TTL value is therefore utilized to distinguish fake IP packets. Since the header only provides the final TTL value, this computation presents a problem. Each Operating System (OS) has a different starting TTL value, and the OS for a certain IP address might vary over time. If the attacker accurately calculates the number of hops between the source and victim, the method will provide a false negative if the real packet originates from an unlisted OS. to differentiate between legitimate and assaultive traffic. It is recommended to employ a path fingerprint

technique, in which each packet has a unique path fingerprint. The path fingerprint represents the path that a packet travels to reach its destination. When the route fingerprint is incorrect, the packet is labelled as spoofed. Because the packet arrives at the same subnet and requires computation at the intermediary nodes, the technique cannot identify subnet faking. The TTL, IP Don't Fragment (DF), window size, and total length values of the TCP/IP header fields are used to determine the OS of a packet. These values are used to construct a fingerprint, which is then appended to the packet at the source. The packet is regarded as real if the fingerprint matches at the receiver side; otherwise, it is recognized as a fake packet.

EXISTING METHODOLOGIES:

The Open Flow switch in the SDN architecture quickly forwards the crucial network data. The administration of the forwarding decision, forwarding, and switch traffic data gathering are under the purview of the SDN controller. The flow table is the primary data structure for the forwarding policy management control in the SDN switch. The SDN finds the flow table entries and delivers the packet to one or more interfaces to handle the relevant network traffic. A header field, counters, and actions are present in each entry. The switch's packet routing is built on top of the flow table. Each flow table has a number of flow entries. The elements in the flow table define the rules for data forwarding. The flow table entry structure diagram is shown in Figure 1. The bulk of the attack detection flow diagram is represented by the extraction characteristic values, classifier judgment, and flow state collection, as shown in Figure 2. The Open flow switch receives requests for a flow table from the flow state collection on a regular basis and responds with the needed data. The six-tuple characteristic values matrix is created by the characteristic values extraction, which is primarily responsible for retrieving the characteristic values related to the DDoS attack from the switch flow table. To specify the relationship between the control and data planes, the Open Flow protocol communicates with the controller about how to handle particular packets.

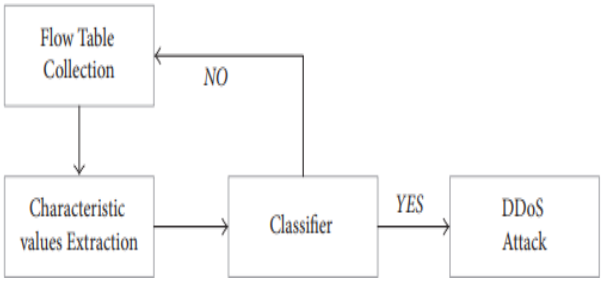


Figure 2: Flow table structure.

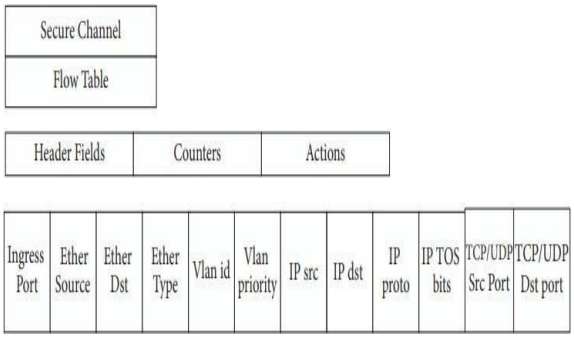


Figure 3: Attack detection process

PROPOSED METHODOLOGY

DDoS attacks are a common threat to the network, despite the fact that the attacker typically has no intention of stealing any data. DDoS attacks essentially aim to deplete system resources to the point where the target can no longer supply their services. The

three subtypes of DDoS attacks include volumetric attacks, protocol attacks, and application layer attacks. A volumetric attack allows an attacker to completely eat up any resources or bandwidth that the victim is using to send traffic to the target. Since a client host might trigger an enquiry from the data plane to the control plane, this kind of attack might also impact the controller and southbound interface in the SDN. Despite the widespread discussion on DDoS attacks in SDN and IoT networks, the abundance of IoT devices and the SDN communication link between controllers and switches still constitute a significant attack possibility. Additional network validations are also required. The SDN's programmability and centralized control also give consumers more options for looking into this issue. In this study, volumetric attack is used.

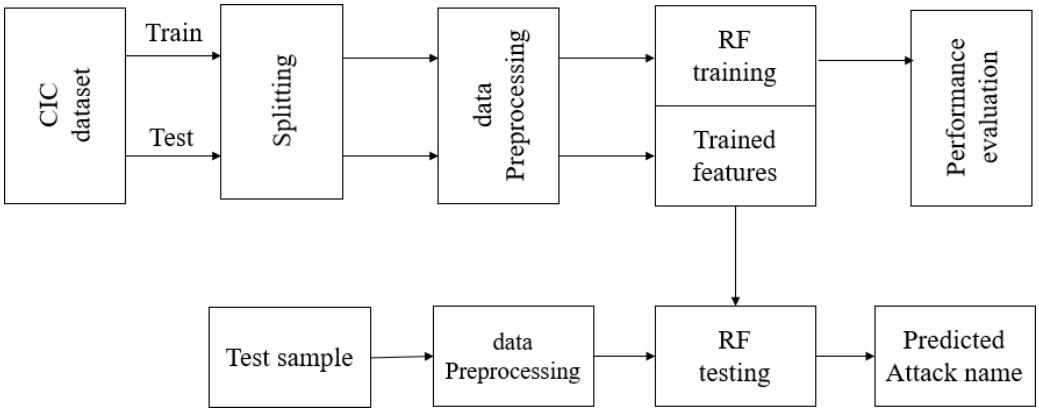


Figure 4 Proposed block diagram

The CIC dataset is initially divided into 20% for testing and 80% for training. The entire dataset is then normalized using a dataset preprocessing operation. Additionally, a random forest classifier is utilized to anticipate DDoS attacks using test data.Performance testing is done to demonstrate the superiority of the suggested approach.

Performance Evaluation CIC dataset:

CICDoS2019's benign and most recent DDoS attacks closely resemble PCAPs from actual real-world data. It also includes the CICFlowMeter-V3 network traffic analysis output, labelled flows based on the time stamp, source and destination IP addresses, source and destination ports, protocols, and attack (CSV files). We used our B-Profile system in the proposed testbed to generate genuine benign background traffic and profile the abstract behaviour of human interactions (Figure 2). We built the abstract behaviour of 25 users for this dataset using the HTTP, HTTPS, FTP, SSH, and email protocols.

Preprocessing:

Data preparation is the process of transforming raw data into something a machine learning model can use. It is the first and most crucial step in the process of creating a machine learning model. When working on a machine learning project, it is not always the case that we are presented with the clean and prepared data. Additionally, you must prepare and clean up your data every time you work with it. So, for this, we employ a data preprocessing activity.

Need of Data Preprocessing:

Since real-world data frequently contains noise, missing values, and may be in an

unfavorable format, it cannot be used to directly train machine learning models. The accuracy and efficiency of a machine learning model are increased by data preprocessing, which is required to clean the data and prepare it for the model.

The dataset can be obtained by importing libraries, importing datasets, searching for missing data, encoding categorical data, dividing a dataset into training and test sets, and feature scaling.

Feature Selection:

One of the fundamental ideas in machine learning, feature selection has a significant impact on the effectiveness of the model. The performance your machine learning models can attain will be greatly impacted by the data characteristics you use to train them. The most crucial phase of creating your layout is feature collection and data cleaning. The process of feature selection involves choosing, either manually or automatically, the features that have the greatest impact on the predictive variable or output you are interested in. The accuracy of your model can be decreased if your data contains irrelevant characteristics, and your model can be trained using irrelevant information.

Model Training:

Run Algorithms: Using this module, we will feed 80% of the training data into Random Forest, XGBOOST, ADABOOST, SVM, Nave Bayes, and KNN algorithms to train a model, which will then be used to test data to measure prediction accuracy.Comparison Graph: We will provide a comparison table and graph of all methods utilising this module.

Predict Attack from Test Data: We will submit test data to this module, and machine learning models will predict attacks based on that data. You can discover test data within the test folder, and this test data contains all features without a class name, which will be predicted by the machine learning model.

Use Case Diagram

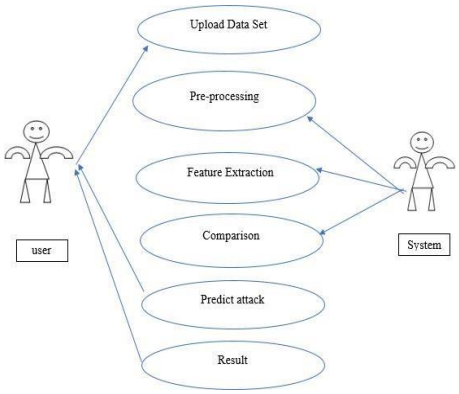


Figure 5.1: Use case diagram

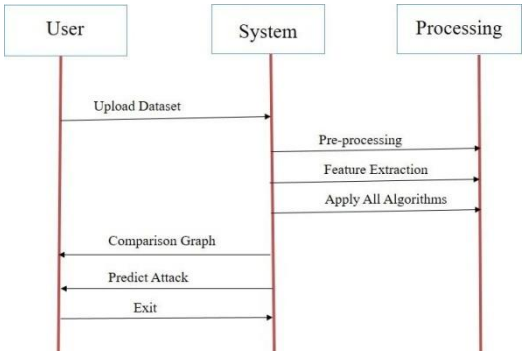


Figure 5.2: Sequence diagram

RESULTS

Test Data you can find inside test folder and this test data contains all featureswithout any classlabel and this label will be predicted by machine learning model.

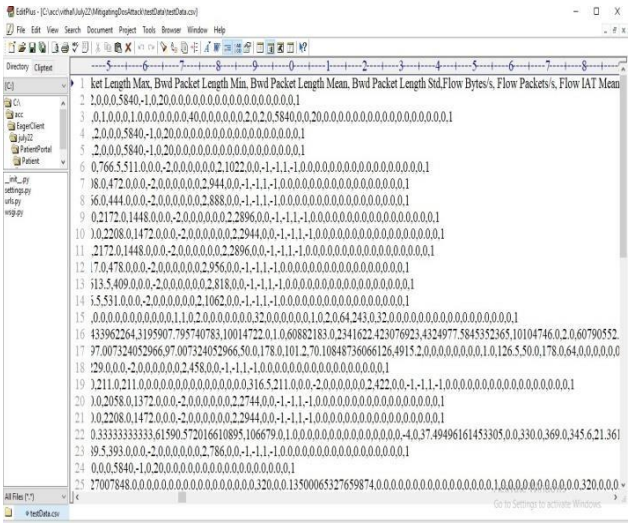


Figure 6.1: Test Data

In above TEST DATA screen there is no class label or attack name and this will be predicted by ML model.

SCREEN SHOTS:

Figure 6.2: Graphical user interface (GUI) Application

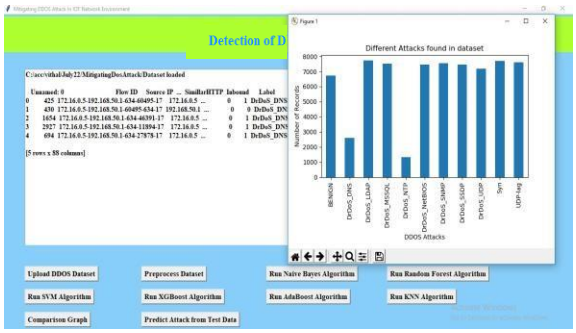


Figure 6.3: Different Attacks found in dataset

Figure 6.4: Preprocessing Dataset Output

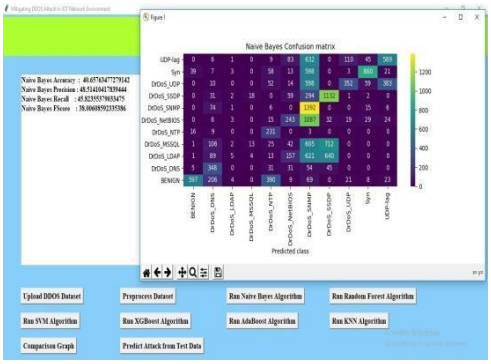
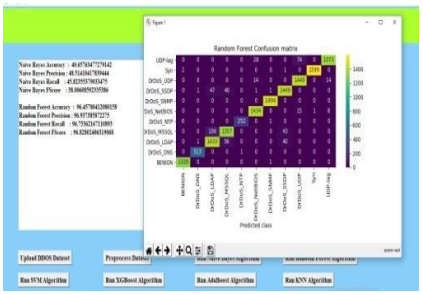


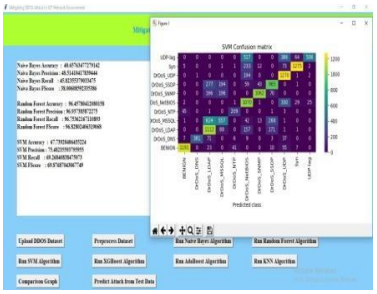
Figure 6.5: Naïve Bayes Confusion

To launch the project, double-click the 'run.bat' file to obtain the screen matrix shown below. In the above screen, click the 'Upload DDOS Dataset' button to upload the dataset and obtain the output shown below.

On the panel up top, where the dataset is loaded, we can see that it contains both numerical and non-numerical data. Attack names are displayed on the x-axis, and the number of such recordings is displayed on the y-axis. After closing the previous graph, select "Preprocess Dataset" to process the dataset and display the screen shown below.The dataset, which has more than 70000 records and 87 features per record, is displayed in its entirety in the screen above. The dataset has been divided into train and test applications, with the training application employing 56685 records for training and 14172 for testing. Once the train and test data are prepared, select "Run Nave Bayes."

In above the screen with Naïve Bayes we got forty percent accuracy and in the confusion matrix graph the x- direction represents predicted classes and the y direction represents TRUE classes and prediction count in same row and column names are the correct prediction and count in different row and column names are the incorrect prediction and we can see Naïve Bayes predicted so many wrong prediction and close above graph and then click on 'Run Random Forest Algorithm' button to get below output. With Random Forest, we achieved greater than 96% accuracy in the image above, and the graph also shows that many of the predictions were accurate. Now that the above graph is closed, click the "Run SVMAlgorithm" button to obtain the output shown below. Close the graph in the above screen after achieving 67% accuracy with SVM, and then click the "Run XGBOOST Algorithm" button to obtain the output shown below.





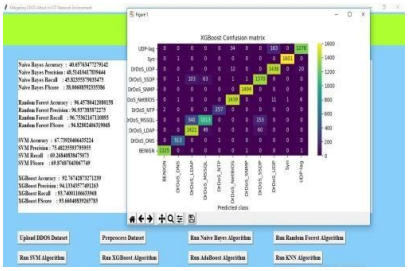
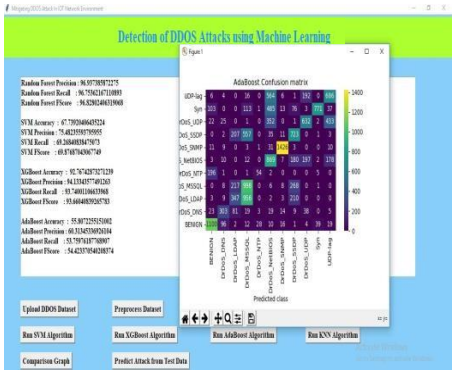


Figure 7.1: Random Forest Confusion matrix Figure 7.2: SVM Confusion matrix Figure 7.3: XGBOOST Confusionmatrix

Close the above graph after achieving 92% accuracy with XGBOOST, and then click the "Run ADA BOOST Algorithm" button to obtain the output shown below .



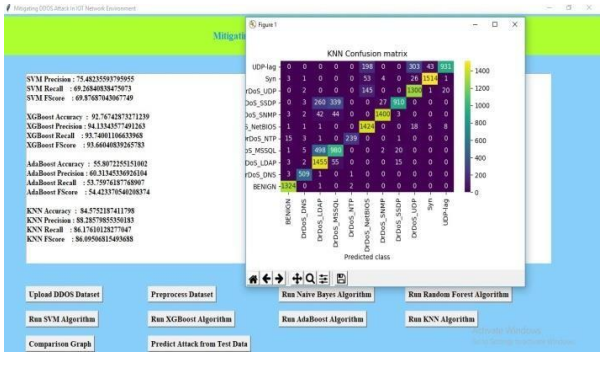


Figure 7.4: ADABOOST Confusion matrix Figure 7.5: KNN confusion matrix

Figure 7.6: Comparison Table

Figure 7.7: Test Data Output

In above

In above screen with ADABOOST we got 55% accuracy and now close above graph andthen click on 'Run KNN Algorithm' button to get below output In above screen with KNN we got 84% accuracy and now close above graph and then clickon 'Comparison Graph' button to get below ggraph and comparison table we can see Random Forest got high accuracy and in above graph different colour bar represents different metrics such as accuracy, precision, recall and FSCORE. Now click on 'Predict Attack from Test Data' button to upload test data and get below output

Selecting and adding the TEST DATA file to the screen above, then clicking the 'Open' button to view the results, will produce. With each individual test record, several attacks and benign (normal) classifications are anticipated in the displays above.

Conclusion and Future Scope

DDoS attacks have become a more significant issue for network security as a result of advancements in networking technologies. It uses commonly used protocols and services while attacking, making it difficult to detect using standard methods. On the basis of the idea of rational reasoning, DDoS attack detection may be modeled as a classification problem that separates "rational" from "irrational" network flow situations. This essay carefully examines the prevalent TCP flood attack, UDP flood attack, and ICMP flood assault. Define the characteristics of data stream information entropy to explain attack behavior. To identify DDoS assaults, it is advised to utilize a random forest classification system. For the aforementioned three kinds of typical attack methods, create categorization models. Finally, through training and learning, it is predicted if the network traffic is normal.

Future Scope

Nowadays, static and dynamic analysis of request data is used to find cyberattacks. Static analysis is based on signatures, and to determine if a packet is normal or contains an attack signature, we compare the new request packet contents with the current attack signature. To find malware or attacks, dynamic analysis will use dynamic program execution, however dynamic analysis takes time. We are using machine learning algorithms to solve this issue and improve the detection accuracy of both old and new malware attacks. These algorithms include Support Vector Machine (SVM), Random Forest, Decision Tree, Naive Bayes, Logistic Regression, K Nearest Neighbors, and Deep Learning Algorithms like Convolution Neural Networks (CNN) and LSTM (Long Short-Term Memory). Various models are among them. When compared to other models, deep learning CNN performed better.

REFERENCES:

[1]. H. Zhang, Z. Cai, Q. Liu, Q. Xiao, Y. Li, and C. F. Cheang, "A surveyon security-aware network measurement in SDN," Security and Communication Networks, Article ID 2459154, 2018.

[2]. J. Cao, M. Xu, Q. Li, K. Sun, Y. Yang, and J. Zheng, "Disrupting SDN via the data plane: a low-rate fow table overow attack," in Proceedings of the 13th EAI International Conference onSecurity and Privacy in Communication Networks, Niagara Falls, Canada, October 2017. 8 Security and Communication Networks

[3]. Z. Cai, Z. Wang, K. Zheng, and J. Cao, "A distributed TCAM coprocessor architecture for integrated longest prefx matching, policy fltering, and content fltering," IEEE Transactions on Computers, vol. 62, no. 3, pp. 417–427, 2013.

[4]. Y. Li, Z. Cai, and H. Xu, "LLMP: exploiting LLDP for latency measurement in sofware-defneddata center networks," Journal of Computer Science and Technology, vol. 33, no. 2, pp. 277– 285, 2018.

[5]. H. Lin and P. Wang, "Implementation of an SDN-based security defense mechanism against DDoS attacks," in Proceedings of the 2016 Joint International Conference on Economics and Management Engineering (ICEME 2016) and International Conference on Economics and Business Management (EBM 2016), Pennsylvania, Penn, USA, 2016.

[6]. J. G. Yang, X. T. Wang, and L. Q. Liu, "Based on trafc and IP entropy characteristics of DDoSattack detection method," Application Research of Computers, vol. 33, no. 4, pp. 1145–1149, 2016.

[7]. A. Saied, R. E. Overill, and T. Radzik, "Detection of known and unknown DDoS attacks usingartifcialneural networks," Neurocomputing, vol. 172, pp. 385–393, 2016.

[8]. R. Braga, E. Mota, and A. Passito, "Lightweight DDoS fooding attack detection using NOX/OpenFlow," in Proceedings of the 35th Annual IEEE Conference on Local Computer Networks (LCN '10), pp. 408–415, Denver, Colo, USA, October 2010.

[9]. N. Z. Bawany, J. A. Shamsi, and K. Salah, "DDoS attack detection and mitigation using SDN: methods, practices, and solutions," Arabian Journal for Science and Engineering, vol. 42, no. 2, pp. 425– 441, 2017.

[10]. X. Wang, M. Chen, C. Xing, and T. Zhang, "Defending DDoS attacks in sofware-defned networking based on legitimate source and destination IP address database," IEICE Transaction on Information and Systems, vol. E99D, no. 4, pp. 850–859, 2016.

[11]. J. Xia, Z. Cai, G. Hu, and M. Xu, "An active defense solution for ARP Spoo ng in OpenFlow network," Chinese Journal of Electronics, vol. 3, 2018.

[12]. Dao, N.N.; Park, J.; Park, M.; Cho, S. A feasible method to combat against DDoS attack in SDN network. In Proceedings of the 2015 International Conference on Information Networking (ICOIN), Siem Reap, Cambodia, 12– 14 January 2015; pp. 309–311. doi:10.1109/ICOIN.2015.7057902.

[13]. Mousavi, S.M.; St-Hilaire, M. Early detection of DDoS attacks against SDN controllers. In Proceedings of the 2015 International Conference on Computing, Networking and Communications (ICNC), Anaheim, CA, USA, 16–19 February 2015; pp. 77–81. doi:10.1109/ICCNC.2015.7069319.

[14]. Dong, P.; Du, X.; Zhang, H.; Xu, T. A detection method for a novel DDoS attack against SDN controllers by vast new low-traffic flows. In Proceedings of the 2016 IEEE International Conference on Communications (ICC), Kuala Lumpur, Malaysia, 23–27 May 2016; pp. 1–6. doi:10.1109/ICC.2016.7510992.