CS584 MACHINE LEARNING: DETECTION OF DDOS ATTACK USING MACHINE LEARNING TECHNIQUE

Sanket Kulkarni
Illinois Institute of Technology
Chicago, USA
skulkarni27@hawk.iit.edu or A20537896

Ameya Hujare
Illinois Institute of Technology
Chicago, USA
ahujare@hawk.iit.edu or A20545367

Deep Pawar
Illinois Institute of Technology
Chicago, USA
dpawar3@hawk.iit.edu or A20545137

***Abstract*—A Distributed Denial of Service (DDoS) attack is an attempt to make a service unavailable by overwhelming the server with malicious traffic. DDoS attacks have become the most tedious and cumbersome issue in the recent past. The number and magnitude of attacks have increased from a few megabytes of data to 100s of terabytes of data these days. Due to the differences in the attack patterns or new types of attacks, it is hard to detect these attacks effectively. In this project, we devise new techniques for causing DDoS attacks and mitigation which are clearly shown to perform much better than the existing techniques. We also categorize DDoS attack techniques as well as the techniques used in their detection and thus attempt an extensive scoping of the DDoS problem. A distributed denial-of-service (DDoS) attack targets websites and servers by disrupting network services. A DDoS attack attempts to exhaust an application's resources. The perpetrators behind these attacks flood a site with errant traffic, resulting in poor website functionality or knocking it offline altogether. Virtualization is used to build and design the hardware components virtually. We propose a machine learning technique which is Random Forest using the decision trees to detect malicious traffic. Our test outcome shows that the Decision Tree algorithm provides better accuracy and detection rate. We train the model using the CICD dataset which compares the datasets by taking realistic network configuration. Complete capture and diverse/multiple attacks and then perform the DDOS attack using virtualization and mitigate it using our ML model. We first simulate a DOS and DDOS attack using Hping and flood the resources of the victim. This data is parsed using a CIC Flowmeter and then run through our ML model for detection of malicious or normal traffic.**

Keywords: Distributed Denial of Service, Random Forest Classifier, SYN, hping, Virtual Machines, Wireshark, CICFlowMeter, hping3

I. INTRODUCTION

Following the information revolution, the industrial environment saw enormous modifications, as evidenced by the rise of concepts such as the smart grid. However, this period of advancement has also brought us new types of cyber risks, with Distributed Denial of Service (DDoS) attacks standing out as a severe threat to vital infrastructure and online activities. In response to these growing threats, there is an urgent need for strong detection techniques capable of combating sophisticated attacks. The danger of Distributed Denial of Service (DDoS) attacks has grown to be a major concern for network security due to the spread of internet-connected devices and the growing reliance on online services. DDoS attacks are designed to stop a targeted server, service, or network from operating normally by flooding it with malicious traffic and making it unavailable to authorized users. These attacks seriously compromise the availability, integrity, and secrecy of online services in addition to causing large financial losses for enterprises. Conventional techniques to identify DDoS attacks frequently depend on signature-based methods, which have limitations in identifying novel attack patterns and variations. More robust and adaptable detection technologies are becoming more and more necessary as DDoS assault strategies advance. Because machine learning algorithms can evaluate vast amounts of network traffic data and find patterns that point to malicious activity, they present a promising method for DDoS detection. Distributed Denial of Service (DDoS) attacks are a danger to the security and reliability of online operations for enterprises worldwide in today's interconnected digital ecosystem. These malicious attacks have the potential to cause serious disruptions, financial losses, and reputational damage. Utilizing machine learning offers a viable way to address this enduring problem. Organizations can improve the resilience of their online services by utilizing machine learning algorithms to detect DDoS attacks in real time. The Random Forest algorithm is one of the many machine learning algorithms that are particularly useful for extracting flow statistics linked to DDoS attacks. It strikes a compromise between resource efficiency and detection accuracy. The goal of this project is to create a reliable framework for detecting DDoS attacks using the Random Forest classifier, specifically designed to handle the special needs of both small- and large-scale network deployments.

A. SURVEY OF RELATED WORK:

DDoS attack prevention has been extensively explored in recent years, with various studies employing different techniques and models. In the survey of related work, several techniques have emerged:

- Neural Networks like CNN, RNN, and GAN for DDoS attack prevention.
- Time-series forecasting models like ARIMA, LSTM, and Prophet have been applied to predict and prevent DDoS attacks by analyzing patterns over time.
- Regression models like Linear, Lasso, Ridge, and Bayesian regression, have been employed in DDoS attack prevention.
- Classification models like Decision Trees, Random Forests, and SVM, have been explored for their effectiveness in classifying network traffic as normal or malicious.

B. How the proposed work is different?

While the existing literature provides valuable insights into DDoS attack detection, the proposed work distinguishes itself in several ways:

- **Algorithmic Approach:** The project adopts a Random Forest algorithm for DDoS attack detection. The algorithm's scalability and parallelization capabilities ensure its efficiency in processing large volumes of data generated during a DDoS attack, contributing to real-time prevention.
- **Real-time Detection:** The primary goal is to develop a model capable of real-time detection, enabling swift responses to mitigate the impact of DDoS attacks as they occur.

The proposed work builds upon the strengths of existing research while introducing novel elements, particularly in the choice of algorithm and the emphasis on real-time adaptability for DDoS attack detection.

C. Milestones:

The proposed work aims to use machine learning for DDoS attack detection, so organizations can better protect their online services and reduce the risk of disruption to their operations.

- **Data collection and pre-processing:** This involves collecting network traffic data and pre-processing it for use in the detection model. The data can be obtained from various sources, such as network traffic logs, packet captures, or sensor data.
- **Model selection and training:** This involves selecting a suitable machine learning algorithm and training it on the pre-processed data to build a detection model. The model can be trained using a supervised or unsupervised learning approach, depending on the availability of labeled data.
- **Model evaluation and validation:** This involves evaluating the performance of the detection model using various metrics such as accuracy, precision, recall, and F1 score.

D. Objectives:

To compare the different Detection techniques for the DOS attack at a low rate and find the appropriate detection technique to mitigate live DDOS attacks. To develop an accurate and reliable model that can effectively detect DDoS attacks in real time, despite the challenges posed by the complexity and variability of network traffic.

II. PROBLEM DESCRIPTION

Real-time detection and mitigation of Distributed Denial of Service (DDoS) attacks is a significant challenge for industrial networks, particularly when low-rate attacks are involved. These attacks can disrupt essential services and compromise the integrity of operations used by overloading network resources with malicious traffic. The main objective of this research is to thoroughly examine and evaluate several detection techniques for both low-frequency Denial of Service (DoS) attacks and DDoS attacks. Our objective is to identify the most effective detection algorithms that can consistently identify the subtle signs of an imminent DDoS attack despite normal network traffic through in-depth study. Furthermore, the project intends to develop a robust and reliable model that can detect DDoS attacks in real-time, despite the inherent complexity and unpredictable nature of network traffic patterns. With a focus on the Random Forest algorithm and advanced machine learning techniques, our objective is to develop a detection model that can adjust to evolving DDoS attack tactics. By achieving these objectives, we aim to strengthen industrial networks' defences against the constant danger caused by DDoS attacks, protecting crucial infrastructure and ensuring uninterrupted operation even in the face of extremely effective cyberattacks.

A. THEORY

- **Random Forest Classifier:** A robust computer learning algorithm used for classification (and regression) tasks is the Random Forest classifier. As an ensemble learning technique, it builds many decision trees during training and outputs the class that represents the mean prediction (regression) or the mode of the classes (classification) of the individual trees. Decision trees, which are a set of inquiries intended to divide data into progressively smaller categories, are the fundamental component of Random Forest. Each question in a decision tree represents a "split" on a specific feature, designed to separate the data in a way that maximizes the distinction among the resulting groups according to the target variable.
- **How the Random Forest Classifier works:**
  - **Bootstrap Aggregating (Bagging):** The foundation of Random Forest is bootstrap aggregating, often known as bagging, which is a technique for creating various training subsets from the original dataset. Different training data subsets are produced through sampling with replacement, and these subsets are then each utilized to train a distinct decision tree.
  - **Random Feature Selection:** Random Forest adds extra randomization into the construction of each tree. It looks for the best feature among a random selection of features compared to looking for the best feature when splitting at a node. Because of the diversity that arises from this, models get typically better.

- **Building Multiple Trees:** It constructs many such trees, and each tree is built to the maximum extent possible. Because there is not the usual pruning that occurs in a single decision tree, each tree is deep and complex.
- **Aggregation:** In a classification problem, every tree in the forest casts a vote for a class; the class predicted by the model is the one with the greatest number of votes, determined by a simple majority. It averages the results of various trees for regression scenarios.

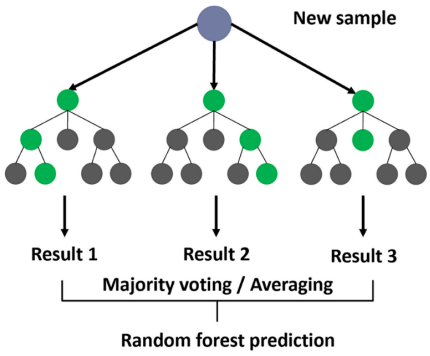


Fig. 1. Random Forest Classifier

- **Significant features:**
  - **Robustness:** Random Forest is more robust than individual decision trees because of the law of large numbers, which reduces variance and reduces overfitting by combining the output of multiple trees, each trained on a slightly different data sample.
  - **Handles Overfitting:** Random Forest's ensemble approach allows for improved generalization for new data, even while individual trees are prone to overfitting to their training set.
  - **Feature Importance:** The significance of each feature in the prediction process can be determined through Random Forest, which can manage a huge number of features. Because they contribute to the greater splits, features selected at the top of trees are typically more significant than traits selected at lower levels.

- **How the Random Forest Classifier identifies DDoS assaults:** Several crucial features make the Random Forest classifier extremely successful at identifying Distributed Denial of Service (DDoS) assaults.
  - **Handling High-Dimensional Data:** In network security, where each feature may hold significant information about network activity, Random Forest's ability to handle huge datasets with multiple input variables without variable loss is essential.
  - **Feature Importance:** Random Forest assigns a numerical value to each feature in order of significance for the classification outcome. This is especially helpful for DDoS detection because it makes it easier to identify the key signs of attack traffic, including IP addresses, certain packet rates, or protocols used.
  - **Adaptability to New Patterns:** Random Forest models are more flexible in responding to changing attack techniques that may not precisely match the patterns of past attack data since they are based on a collection of decision trees rather than a single model.
  - **Robustness to Noise:** Handling noisy and unbalanced data is a common task for DDoS detection. Because Random Forest is naturally resistant to noise, it is more dependable in these kinds of settings.
  - **Efficiency in Training and Prediction:** Random Forest is comparatively efficient in both the training and prediction phases, which is crucial for real-time DDoS attack detection in dynamic network environments, even though ensemble models can be computationally demanding.

B. APPLICATION

- **Methodologies used in Model:**
  1) **TCP Three-Way Handshake:** A three-way handshake is started by the client when it wishes to connect to a server over TCP. The server receives a SYN-ACK (synchronize-acknowledge) packet from the client after it receives a SYN (synchronize) packet. To finish the handshake and establish the connection, the client subsequently sends an ACK (acknowledge) packet.
  2) **Attack Initialization:** In a SYN flood attack, the attacker bombards the target server with SYN packets in large numbers while remaining silent in response to the server's SYN-ACK messages. These SYN packets contain spoof source IP addresses from the attacker, which makes it challenging for the server to distinguish between malicious and legitimate connection attempts.
  3) **Resource Exhaustion:** The target server allocates system resources to handle each incoming SYN packet and reserves a TCP connection slot for each connection request. However, since the attacker does not complete the handshake process by sending the final ACK packet, the server keeps waiting for a response that never arrives. As a result, the server's resources, such as memory and processing power, are gradually consumed and become exhausted.
  4) **Denial of Service:** The server is unable to handle valid connection requests from actual clients because its resources are overloaded. A denial of service occurs as a result, causing the server to become slow or unresponsive, with the possibility of a server failure or unavailability.

- **Hping3:** Similar to how ping displays ICMP replies, Hping3 is a network utility that can send customized ICMP/UDP/TCP packets and show target replies. We can use several protocols to test network performance, conduct (spoof) port scanning, and check firewall rules. Among its numerous capabilities include the ability to send files between a covered channel, a traceroute mode, support for TCP, UDP, ICMP, and RAW-IP protocols, and many more. With the hping3 program, you can send packets that have been altered in terms of size, quantity, and fragmentation to take down the target and go around or breach firewalls.

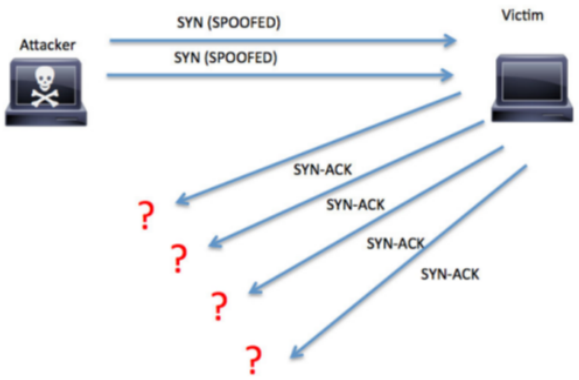


Fig. 2. SYN attack using Hping3

- **Virtualization:** Rather than using a physical environment, virtualization generates a simulated, or virtual, computing environment. To simulate an hping DDoS attack using virtualization, you can follow these general steps:
  1) **Set up a Virtualization Environment:** Install a virtualization platform on your host computer, such as KVM, VMware, or VirtualBox.
  2) **Create Virtual Machines (VMs)**: To replicate the attack scenario, build two or more virtual machines. The attacker will be one virtual machine (VM) that is executing the hping tool, and the target(s) of the attack will be the other.
  3) **Networking Configuration:** To create a connection between the attacker's virtual machine and the target virtual machine, configure the virtual network settings. Depending on your needs, you can employ host-only, NAT, or bridged networking.
  4) **Install Operating Systems:** Set up the target and the attacker's preferred operating systems on the virtual computers. Make sure you have correctly configured the required networking settings (IP addresses, subnet masks, and gateways).
  5) **Install hping:** Set up hping on the virtual machine of the attacker. Depending on the operating system you are running, you may download hping using package managers like apt-get or yum or from the official website.
  6) **Configure the Attack:** Establish the hping DDoS assault's parameters, including the target IP address, port number, attack type (such as SYN or UDP flooding), and attack rate.
  7) **Initiate the Attack:** To begin a DDoS attack against the target VM(s), run the hping command on the attacker's virtual machine with the selected parameters. To determine the effect of the attack, keep focus on the target(s)'s system performance and network traffic.
- **Wireshark:** An open-source, free packet analyzer is called Wireshark. It is used in analysis, software development, and communications protocol troubleshooting on networks. A graphical user interface (GUI) in Wireshark, formerly known as Ethereal, allows users to visualize network traffic across a variety of networks, including Ethernet, Wi-Fi, and others. Users of Wireshark can read packets from a previously saved capture file or record current network data. The tool helps in the examination of each packet's various fields by displaying the contents of each packet in detail and breaking them down into a protocol hierarchy. Understanding network behavior and identifying problems like packet loss, illegal access, and network congestion can be significantly helped by this. Additionally, Wireshark has strong filtering features that let users isolate traffic based on parameters like IP address, protocol type, or port number, which facilitates the analysis of network traffic elements.

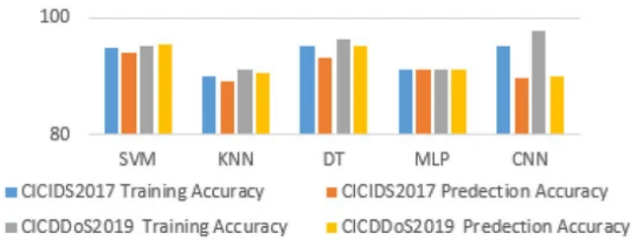
- **The Dataset: CICIDS2017 Dataset** The Canadian Institute for Cybersecurity Intrusion Detection Systems 2017 Dataset, or "CICIDS 2017 Dataset," is the dataset we are using in this project. It is an extensive collection of network traffic data that was obtained while researching to create intrusion detection systems that are effective. The dataset is described in detail below:
  - **Data Source:** The dataset consists of network traffic captured in a research network environment designed to mimic the topology of a real-world corporate network.
  - **Features:** The dataset contains many features that have been taken from network traffic, including attributes at the packet and flow levels. A few of the features that are available in the dataset are as follows:
    * Source and destination IP addresses
    * Source and destination port numbers
    * Protocol (e.g., TCP, UDP)
    * Packet size
    * Flags (e.g., SYN, ACK)
    * Flow duration
    * Number of packets in the Flow

- Total bytes transferred
- Label indicating whether the traffic is benign or malicious (i.e., normal or an attack)

- **CICFlowMeter:** CICFlowMeter is a software application utilized to analyze and monitor network traffic. The abbreviation for this is "Communication Information Collector Flow Meter." In short, it records network traffic data and offers information on a range of topics, including source and destination IP addresses, traffic volume, and protocols being used, among many other things. In cybersecurity, this type of tool is frequently used to find anomalies, pinpoint possible dangers, and enhance network efficiency. It aids in the better understanding and management of network infrastructure by analysts and administrators. In our machine learning model, we are using CICFlowMeter to convert pcap (Packet Capture) files into CSV (Comma Separated Values) files.

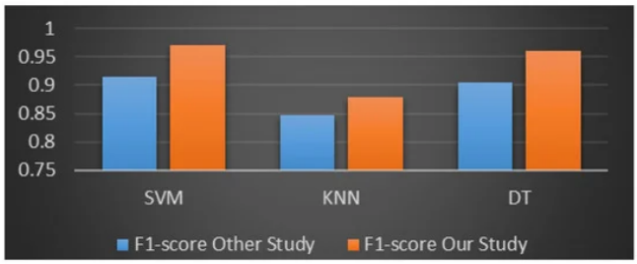
III. COMPARATIVE ANALYSIS OF DIFFERENT MACHINE LEARNING MODELS

- **Accuracy Prediction:**



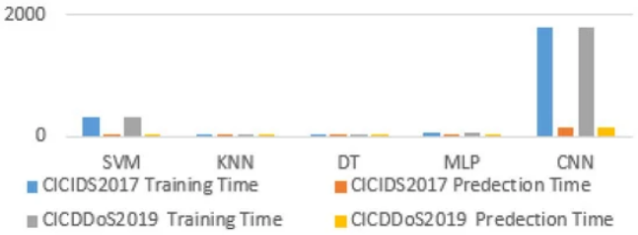
After analysis, we found that the Decision Tree (Random Forest) provides comparatively high training and prediction accuracy while working on CICDS2017 and CICDDoS2019 datasets.

- **F1-score study:**



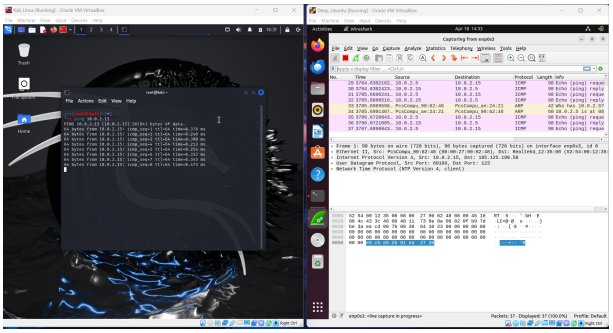
After analysis, we found that the Decision Tree (Random Forest) provides a 90-95 percent F1 score while comparing it with SVM and KNN models.

- **Training and prediction time:**
After analysis, we found that the Decision Tree (Random Forest) provides comparatively less training and prediction time while working on CICDS2017 and CICDDoS2019 datasets.



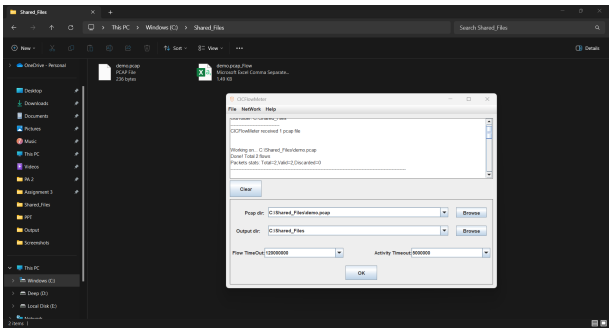
IV. STEPS FOLLOWED TO EXECUTE THE MODEL

1) Sending normal ICMP packets from Kali Linux to Ubuntu Linux and capturing them using Wireshark.

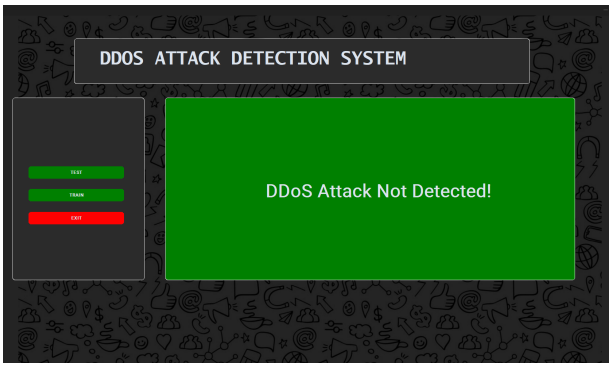


2) Verifying the incoming packets using system monitoring.

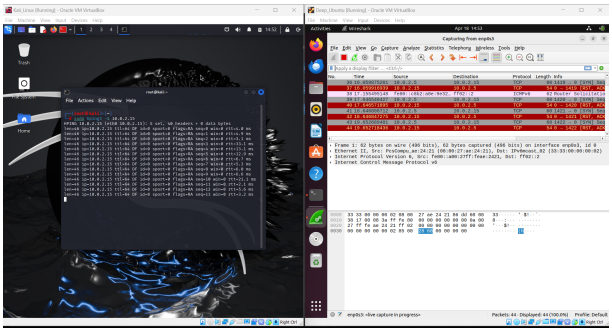
3) Converting captured packets from .pcap files to .csv files using CICFlowMeter.



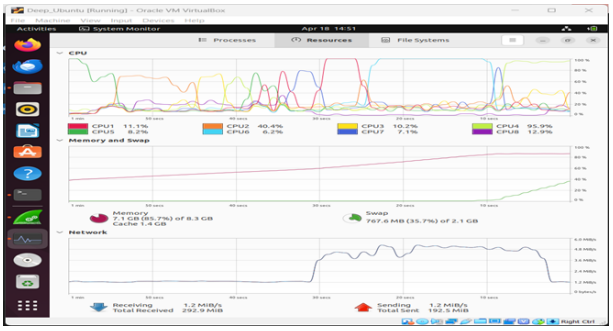
4) After uploading the generated CSV file into the system, we can successfully detect that there is no DDoS attack is detected.



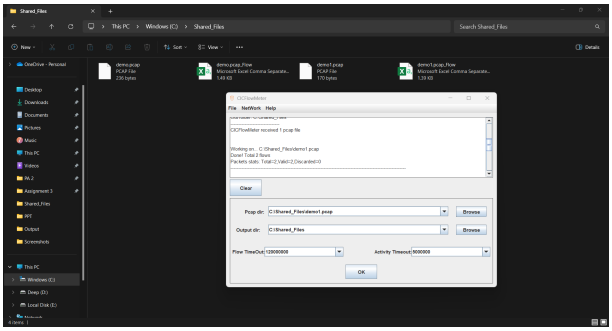
5) Similarly, to check the DDoS attack we can send malicious packets using the Hping command as shown below:



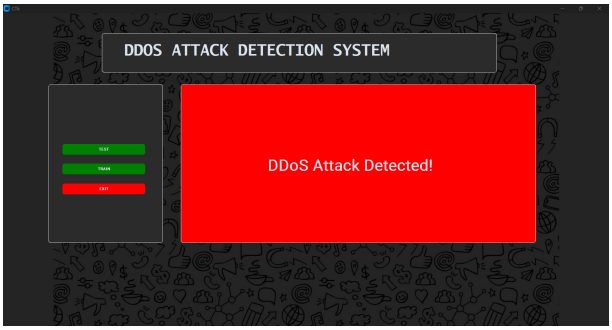
6) Verifying the incoming packets using system monitoring.



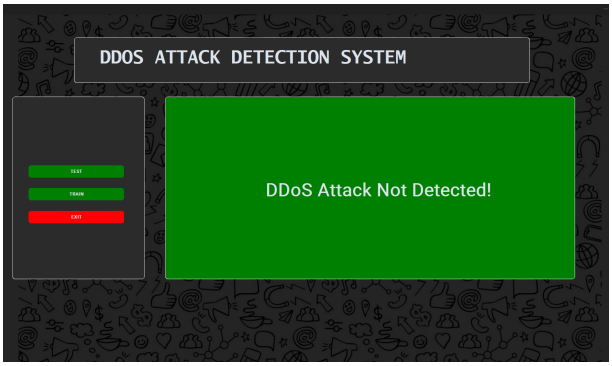
7) Converting the newly captured packets from .pcap files to .csv files using CICFlowMeter.



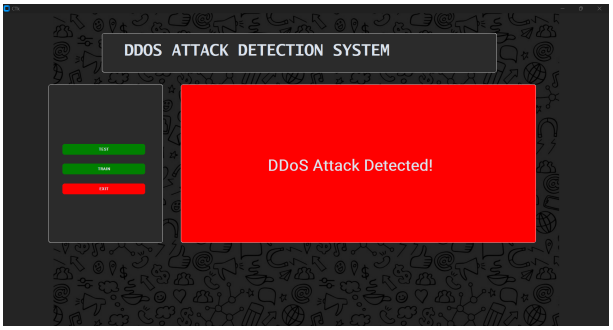
8) After uploading the generated CSV file into the system we can detect that the DDoS attack in the packets is successfully detected.



V. RESULTS

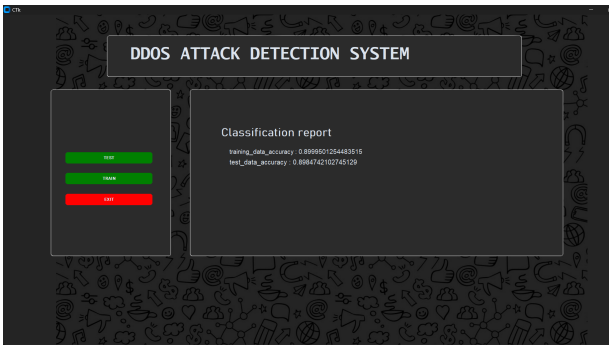


Here, Wireshark was used to carefully watch and record the flow of regular ICMP packets as they were transferred from a Kali Linux system to an Ubuntu Linux machine in our experiment. With the use of a specialized system monitoring tool, this configuration made it possible to thoroughly verify incoming packets, guaranteeing their integrity and expected behaviour throughout packet transmission. Subsequently, the captured packets, initially stored in. pcap format, were converted into .csv files utilizing CICFlowMeter for more refined analysis. After we uploaded these CSV files, our machine learning model effectively examined the data and verified that there had been no DDoS attack impact. This outcome demonstrates how well the model can differentiate between typical network traffic and possible cybersecurity threats.



Here, malicious packets were simulated using the Hping command, which effectively generated a type of network traffic common to DDoS scenarios, to assess the model's capacity to recognize DDoS attacks. The accuracy of our test circumstances was ensured by transmitting these packets and then verifying that they were received through a system monitoring tool. To enable in-depth analysis, the malicious packets, which were originally recorded in. pcap format, were transformed into .csv files using CICFlowMeter. The model effectively identified the existence of a DDoS attack after uploading these CSV files into our system, demonstrating its ability to identify between benign and malicious network activity.

To check the classification report we can click on the TRAIN button which gives us around 89 percent training and testing accuracy which suggests that the model correctly predicted 89 out of every 100 examples in the training and testing dataset.



VI. CONCLUSION

The Random Forest algorithm proves to be a highly efficient technique for identifying DDoS attacks and has several advantages that are vital for strengthening network security. Random Forest makes it easier to classify network traffic instances accurately by exhibiting robustness against noise and outliers, handling high-dimensional data, and capturing complicated patterns. This allows it to distinguish between DDoS attacks and regular behavior with accuracy. Additionally, network managers can benefit from its feature importance analysis, which helps them identify key DDoS attack signs and improve security protocols accordingly. Furthermore, Random Forest can manage massive data sets and adjust to changing attack patterns because of its scalability and flexibility, which guarantees rapid mitigation and response measures. It is crucial to recognize that although Random Forest offers a strong method, it is not a cure-all for DDoS assault detection. The quality of the dataset, feature engineering, parameter adjustment, and ongoing monitoring and improvement are all critical to the model's efficiency. To stay accurate and flexible in response to increasing attack methods, regular reviews and upgrades are necessary. In conclusion, the study emphasizes how important it is to use the Random Forest algorithm to strengthen network security infrastructure, identify possible risks instantly, and efficiently lessen the effects of DDoS attacks. In the field of cyber threat identification, this algorithm's analysis and use have yielded important insights regarding the significance of data quality, model adaptability, and ongoing enhancement. Future efforts to strengthen network security and resilience against changing cyber threats will be driven by these lessons. It is crucial to remember that unless you have the appropriate authorization and are running these simulations for educational or scientific purposes, carrying out DDoS attacks is prohibited and unethical. Always make sure you are operating within the bounds of the law and acceptable usage standards, and that you have permission to do such tests.

VII. LIMITATIONS

1) **Real-Time Detection:** Although the model may work well with historical data, real-time DDoS attack detection can be more difficult because quick processing and decision-making are required. The amount of processing power available or the amount of time needed to capture and prepare packet data could be limiting factors.
2) **Scalability:** The system must be able to manage higher data volumes as network traffic grows. This may put a strain on data storage and processing speed, necessitating the use of more capable hardware or software that is tuned.
3) **Adaptation to New Attacks:** Cybercriminals are always changing their strategies. If a trained model is not constantly updated with new training data, its static nature may prevent it from quickly adapting to new or modified DDoS methods of attack.
4) **Resource-intensive:** Processing big datasets with a machine learning model after using tools like Wireshark and CICFlowMeter might be resource-intensive. This could prevent our system from being deployed in settings with limited computational resources.
5) **Complexity of Deployment:** There are multiple steps in the setup process, from packet capturing to packet classification, and each one needs to be properly configured and maintained. Maintenance and deployment may be challenging due to this complexity.

VIII. FUTURE WORK

Several areas could use enhancement in the context of DDoS attack detection models that employ the Random Forest algorithm. The following are some possible directions for further study and advancement:

- **Improvements to Real-Time Processing:** Timely DDoS detection depends on the system's ability to handle real-time data analysis. This can involve developing stream processing tools that enable data analysis to happen practically as soon as it is captured. Moreover, the latency associated with data processing and model execution can be greatly decreased by utilizing hardware accelerations like GPUs or FPGAs.

- **Enhancements to Scalability:** To address scalability, distributed computing approaches are used through the use of platforms such as Apache Spark or Hadoop, which are built to efficiently process massive amounts of data across numerous computer resources. Adopting cloud-based analytics could also aid in scaling management by making use of the adaptable and expandable resources provided by cloud services.

- **Adaptive Learning:** By using incremental learning, the model may continuously adjust to new attack patterns and data without requiring total retraining. By establishing a feedback mechanism, the system can improve its detection capabilities dynamically depending on feedback from the actual world and learn from its operational experiences.

- **Resource Optimization:** The system can be made more effective by reducing the computational and memory needs of the detecting algorithms. In order to facilitate quicker and more efficient data processing, it is also important to put effective data management strategies into place to guarantee that massive amounts of network traffic data are handled and stored effectively.

- **Simplifying Deployment:** Deployment can be made simpler and less complicated by creating automated tools for system setup and maintenance. Standardized, modular components can make upgrades and maintenance faster, streamline processes, and lower the chance of errors for routinely performed functions in the system.

IX. CONTRIBUTION TABLE

Student Name	A#	Project Contribution
Sanket Kulkarni	A20537896	33.33%
Ameya Hujare	A20545367	33.33%
Deep Pawar	A20545137	33.33%

REFERENCES

[1] S. S. Panwar, Y. P. Raiwani and L. S. Panwar, "An Intrusion Detection Model for CICIDS-2017 Dataset Using Machine Learning Algorithms," 2022 International Conference on Advances in Computing, Communication and Materials (ICACCM), Dehradun, India, 2022, pp. 1-10, doi: 10.1109/ICACCM56405.2022.10009400.

[2] P. S. Saini, S. Behal and S. Bhatia, "Detection of DDoS Attacks using Machine Learning Algorithms," 2020 7th International Conference on Computing for Sustainable Global Development (INDIA.Com), New Delhi, India, 2020, pp. 16-21, doi: 10.23919/INDIACom49435.2020.9083716.

[3] S. Peneti and H. E, "DDOS Attack Identification using Machine Learning Techniques," 2021 International Conference on Computer Communication and Informatics (ICCCI), Coimbatore, India, 2021, pp. 1-5, doi: 10.1109/ICCCI50826.2021.9402441.

[4] S. Santhosh, M. Sambath and J. Thangakumar, "Detection Of DDOS Attack using Machine Learning Models," 2023 International Conference on Networking and Communications (ICNWC), Chennai, India, 2023, pp. 1-6, doi: 10.1109/ICNWC57852.2023.10127537.

[5] G. Ajeetha and G. Madhu Priya, "Machine Learning Based DDoS Attack Detection," 2019 Innovations in Power and Advanced Computing Technologies (i-PACT), Vellore, India, 2019, pp. 1-5, doi: 10.1109/i-PACT44901.2019.8959961

X. GITHUB LINK

Link: https://github.com/DeePawar28/detection-of-ddos-attack-using-random-forest-classifier