Development of Cyber Attack Detection System in Web Applications

Project Report

Submitted By:

Gautam Peluru (19BCS085)

Rutwik Pasumarthi (19BCS084)

Satwik Pasumarthi (19BCS083)

Dinesh Kothapalli (19BCS060)

Under the Guidance of,

Dr. Malay Kumar

Assistant Professor
Department of Computer Science & Engineering

INDIAN INSTITUTE OF INFORMATION TECHNOLOGY, DHARWAD

CERTIFICATE

It is to certify that the work contained in the project report titled **"Development of Cyber Attack Detection System in Web Applications**" by
Peluru Gautam(19BCS085), Pasumarthi Satwik(19BCS083), Pasumarthi Rutwik(19BCS084), and Dinesh Kothapalli(19BCS060) have been submitted for the fulfillment of the requirement for the degree of "Bachelor of Technology in Computer Science & Engineering". Their work has been found satisfactory and hereby approved for submission.

Signature of the Supervisor

Dr. Malay Kumar

Assistant Professor

Department of Computer Science and Engineering IIIT-Dharwad

DECLARATION

We hereby declare that the written project submission presented here is a genuine and original representation of our ideas, and we have not taken it from any other source. Any resources used by other authors have been appropriately credited through citations. We confirm that we have followed the principles of academic honesty and integrity in conducting this work, with utmost sincerity and respect towards our profession and the institute. We acknowledge the serious consequences of any falsification of information or misinterpretation of data and figures. Therefore, we have made every effort to produce the most accurate and reliable response and results possible. This declaration serves as our assurance of the authenticity and integrity of this project submission.

Satwik Pasumarthi(19BCS083)
Gautam Peluru(19BCS085)
Rutwik Pasumarthi(19BCS084)
Dinesh Kothapalli(19BCS060)

APPROVAL SHEET

The project entitled “Development of Cyber Attack Detection System in Web Applications” by Satwik Pasumarthi(19BCS083), Gautam Peluru(19BCS085), Rutwik Pasumarthi(19BCS084), and Dinesh Kothapalli(19BCS060) is approved for the degree of Bachelor of Technology in Computer Science & Engineering and Bachelor of Technology in Electronics and Communication Engineering.

Signature of the Supervisor

Dr. Malay Kumar

Assistant Professor

Department of Computer Science and Engineering IIIT-Dharwad

Head of Department

Dr. Pavan Kumar C

Assistant Professor

Department of Computer Science and Engineering IIIT-Dharwad

ABSTRACT

The ubiquitous nature of the internet and the widespread use of web applications have made them an attractive target for cyber-attacks. Among these attacks, Distributed Denial of Service (DDoS) attacks pose a significant threat to users and enterprises. Therefore, effective DDoS detection methods are essential to reduce the possible damage caused by such attacks. This project report explores the usage of machine learning (ml)-based algorithms and entropy-based detection for DDoS detection.

The project's methodology involves using ml-based algorithms to analyze network data and identify anomalous behavior patterns that indicate the presence of a DDoS attack. Entropy-based detection is also utilized to detect odd data patterns in network traffic. The effectiveness of these methods was evaluated using real-world data, and the results showed a high level of accuracy in recognizing various forms of DDoS assaults.

The findings of this project report highlight the potential of ml-based algorithms and entropy-based detection for DDoS detection. The use of these techniques can significantly reduce the risks posed by DDoS attacks to users and enterprises. However, further study in this field is necessary to increase the efficacy of cyber security measures.

This project report is significant as it contributes to the development of effective DDoS detection methods that can be integrated into cyber security measures. The use of ml-based algorithms and entropy-based detection can enhance the accuracy and reliability of DDoS detection systems. This project report's findings can be useful to cyber security professionals, researchers, and policymakers interested in improving the security of web applications and networks.

Table of Contents

CHAPTERS **Page No**

CHAPTER 1

INTRODUCTION:

Web applications have become an essential aspect of modern life, with millions of people relying on them to do a variety of activities. Web apps are used to access and exchange sensitive information, from social networking to online banking and shopping. Because of the increased reliance on online applications, the risk of cyber assaults targeting them has increased dramatically.

Cyber threats are growing more common and sophisticated, posing a substantial risk to both organizations and individuals. Websites are especially vulnerable to assaults because they are widely accessible and frequently include sensitive information. Cybercriminals can use website vulnerabilities to obtain access to personal data, implant malware, or bring the entire site down, inflicting substantial harm to persons and businesses.

As a result, recognizing and mitigating cyber threats is crucial for safeguarding websites and their users. The need of creating appropriate security measures to identify and prevent cyber assaults on websites will be discussed in this article. It will also look at the many sorts of assaults that can occur, as well as the potential ramifications of such attacks. New detection and mitigation techniques for these cyber threats are desperately needed.

A DDoS assault, also known as a Distributed Denial of Service attack, is a sort of cyber attack whereby multiple systems or devices are utilized to overload a website or network with traffic, rendering it inaccessible to users. DDoS attacks often include attackers infiltrating a huge number of devices with malware, transforming them into bots or zombies that can be controlled remotely. The attackers then instruct these bots to flood the targeted website with visitors, overloading its resources and causing it to crash.

DDoS attacks may be devastating to websites and their resources. These can result in major income loss for websites, harm to a company's image, or even the entire shutdown of a site. Moreover, DDoS attacks can have an influence on the user experience, making genuine users' access to the site or services provided difficult or impossible. Additionally, DDoS assaults are frequently employed as a diversion method, diverting security teams' attention away from more focused and harmful operations.

Overall, DDoS attacks pose a severe danger to websites and their resources, and strong security measures to identify and prevent such attacks are critical. To

successfully detect and prevent such attacks, powerful security mechanisms that can detect odd traffic patterns and take appropriate action to limit the effects of such attacks are required. Machine learning (ML) algorithms and statistical techniques are some of the methodologies for developing such security measures.

Machine learning techniques may be used to analyze network data and identify patterns that indicate a DDoS attack. The ML system can learn to discriminate between regular network traffic and DDoS assaults by training it with data on both. Statistical approaches, on the other hand, may be used to detect odd data patterns in network traffic. These approaches examine numerous network traffic parameters, such as packet size and timing, to find patterns that indicate a DDoS assault.

This project is mainly concerned with the detection and mitigation of DDoS assaults on websites. The majority of the research papers we referenced are concerned with identifying and mitigating attacks on Software-defined networks (SDNs), and they have adopted numerous strategies to do so. Some use statistical approaches, such as assessing the entropy of a packet received from a known IP address, while others use modern-age technology, such as machine learning and deep learning, to distinguish between genuine and fraudulent requests.

This paper presents the use of both ML-based algorithms and statistical methods to build a system for detecting DDoS attacks. The system can successfully identify and mitigate DDoS assaults while avoiding false positives by integrating the capabilities of both techniques. The article will go through the system's implementation, its performance in identifying DDoS assaults, and the possible advantages of employing a hybrid method for DDoS detection.

❖ PROBLEM FORMULATION:

The rise in cyber-attacks on web applications poses a significant threat to the confidentiality, integrity, and availability of sensitive information. Current security measures, such as firewalls and intrusion detection systems, are insufficient in detecting and mitigating sophisticated cyber-attacks that exploit web application vulnerabilities. To address this issue, a robust and effective cyber-attack detection system specifically designed for web applications is necessary to proactively identify and respond to various types of attacks, including SQL injection, cross-site scripting (XSS), and cross-site request forgery (CSRF). The aim of this research is to create a comprehensive cyber-attack detection system for web applications that utilizes advanced

machine learning algorithms, anomaly detection techniques, and behavioral analysis to accurately detect and notify cyber-attacks in real-time. This will enhance the security of web applications and protect sensitive data from unauthorized access and exploitation.

CHAPTER 2
BACKGROUND STUDY:

❖ Existing Machine learning-based detection:

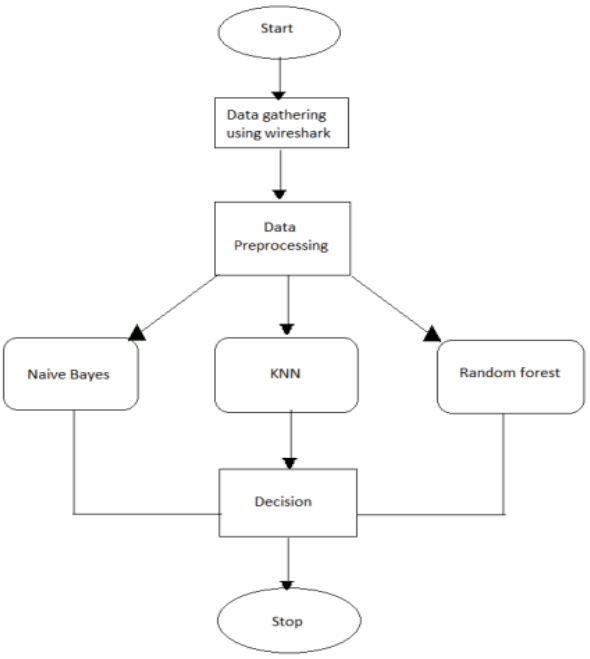

Fig1: Workflow of an ML-based detection system [1]

- **Data Pre-processing**

Pre-processing data is crucial for enhancing the algorithm's performance, and it involves eliminating noise and converting data into numerical values. Feature extraction is another important phase of data pre-processing, where useful information is extracted from raw data to facilitate better learning and prediction by the algorithm.

- **Feature Extraction**

Feature extraction is a crucial aspect of data classification, and for this project, three features have been identified, specifically packet length, delta time, and protocol.

- **Data Modification And Classification**

The features that have been identified are then converted into a numerical format, and these numerical representations are used as input for the ML models. The dataset is split into training data (30%) and test data (70%), and all ML models are processed concurrently for accuracy evaluation..

- **Decision Making**

The accuracy of the model is significantly improved since the final decision is based on a majority vote among all the algorithms used. This project has employed three distinct methods, which have resulted in three separate outcome nodes. By using a voting mechanism, the best outcome is selected from these nodes, which further enhances the accuracy of the model.

- **Random Forest**

Random Forest is an ensemble learning algorithm utilized for various machine learning tasks, including classification and regression. It is based on the decision tree algorithm, a supervised learning technique suitable for classification and regression problems.

The Random Forest algorithm generates numerous decision trees, each of which is trained on a subset of the data and a random set of features. The predictions of these trees are then combined to produce the final prediction. For classification tasks, this combination is achieved by taking a majority vote, while for regression tasks, it is obtained by calculating an average value.

The use of multiple decision trees and random subsets of features helps to improve the accuracy and stability of the predictions made by the model. By creating a diverse set of decision trees that are trained on different subsets of the data, the model can generalize better to new data and avoid overfitting.

- **XG Boost**

XGBoost, which stands for Extreme Gradient Boosting, is a widely-used machine learning algorithm for classification, regression, and other tasks. It is a form of gradient boosting, an ensemble learning technique that trains a sequence of weak learners, such as decision trees, on the residual errors of preceding learners.
XGBoost is a powerful and efficient implementation of gradient boosting that is designed to scale to large datasets and achieve high prediction accuracy. It uses a combination of regularization techniques to prevent overfitting and improve generalization. It also includes advanced features such as tree pruning, early stopping, and parallel processing to further improve performance.

- **Multi-Layer Perceptron**

A Multi-layer Perceptron (MLP) is a type of artificial neural network utilized in machine learning to classify and predict data. This network is designed to be feedforward, meaning that the data inputs only move in one direction through the network and do not create any cycles or loops.
The MLP is composed of multiple layers of neurons, each layer performing a specific computation. The first layer is the input layer, which takes the input data and passes it to the hidden layers. The hidden layers are composed of one or more layers of neurons that apply non-linear transformations to the input data. The output layer is the final layer of neurons that produces the output of the network.
Each neuron in the MLP is connected to the neurons in the previous and next layers by a set of weights, which are adjusted during the training process to minimize the error between the predicted output and the actual output. The weights are adjusted using a backpropagation algorithm, which calculates the error between the predicted and actual output and updates the weights to reduce the error.

- **Hard Voting Ensemble**

Hard voting is a method used in ensemble learning to merge the predictions of several machine learning models to produce a conclusive prediction. In this approach, each model in the ensemble provides its own prediction, and the final prediction is determined by a majority vote based on the individual predictions.

❖ Existing Entropy-based Detection:

Entropy-based detection for DDoS detection is a technique used to detect anomalous traffic patterns by analyzing the randomness or unpredictability of the traffic flow. In this method, the system calculates the entropy of the network traffic in real time and compares it to a predetermined threshold value. If the entropy value exceeds the threshold, it is considered to be an indication of a DDoS attack.

Entropy is a measure of the randomness or unpredictability of a system. In the context of network traffic, entropy can be used to analyze the diversity of the source and destination IP addresses, the protocols used, the packet size distribution, and other characteristics of the traffic flow. In a normal network, the traffic flow is expected to have a certain level of entropy, which can be used as a baseline value for comparison.

During a DDoS attack, the traffic flow becomes more predictable, as it is generated by a botnet or other automated tool. This can result in a decrease in the entropy of the traffic flow, which can be detected by the system. By comparing the real-time entropy value to the baseline value, the system can determine if the traffic flow is anomalous and potentially indicative of a DDoS attack.

This algorithm aims to detect Distributed Denial of Service (DDoS) attacks in a network by calculating the entropy of the traffic flow. The algorithm consists of three functions: ENTROPY, COLLECT_STATISTICS, and ENTROPY_DETECTION.

The ENTROPY function collects traffic flow from a switch and stores it in a file.

The COLLECT_STATISTICS function calculates the probability of each destination IP address in the current interval and then calculates the entropy of the network using the probability values. The difference between the calculated entropy and the normal entropy value is also calculated.

The ENTROPY_DETECTION function compares the calculated difference with a predefined threshold value. If the difference is greater than the threshold, the DDoS detected count is incremented.

If the DDoS detected count is greater than the minimum DDoS detected in a particular window, an alert of a DDoS attack is generated. If the DDoS detected count is less than the minimum DDoS detected in a particular window, the program counter is incremented.
Overall, this algorithm provides a mechanism to detect DDoS attacks by analyzing the entropy of the network traffic flow.

```
async def calculateEntropy(ip_dst):

                self.pktCnt += 1
                if ip_dst in self.ipList_Dict:
                    self.ipList_Dict[ip_dst] += 1
                else:
                    self.ipList_Dict[ip_dst] = 0

                if self.pktCnt == 5:
                    self.sumEntropy = 0
                    self.ddosDetected = 0

                    for ip_dst,value in self.ipList_Dict.items():
                        prob = abs(value/float(self.pktCnt))
                        if (prob > 0.0):
                            ent = -prob * math.log(prob,2)
                            self.sumEntropy = self.sumEntropy + ent

                    if (self.sumEntropy < 2 and self.sumEntropy != 0):
                        self.counter += 1
                    else:
                        self.counter = 0
                    if self.counter == 10:
                        self.ddosDetected = 1
                        #print("Counter = ", self.counter)

                        if(self.ddosDetected == 1):

                            print("
ENTROPY HAS DETECTED DDOS ATTACK PASSING THE TRAFFIC TO ML MODULE")
                            #ML Model
                        else:
                            pass
                        self.counter = 0
                    self.pktCnt = 0
                    self.dst_ipList = []
                    self.ipList_Dict = {}
```

Fig2: An entropy-based detection algorithm

- This algorithm is implementing a function called "calculate entropy" that detects DDoS attacks based on entropy calculation. The function takes an input parameter "ip_dst" that represents the destination IP address of a network packet.

- The function maintains a dictionary called "ipList_Dict" to store the number of packets received for each unique IP address. The variable "pktCnt" keeps track of the number of packets processed so far.

- When a new packet is received, the function increments "pktCnt" and checks if the destination IP address already exists in the "ipList_Dict" dictionary. If it does, the count for that IP address is incremented. If not, a new entry is created with a count of 0.

- The function then checks if "pktCnt" has reached a threshold value of 5. If so, the entropy is calculated based on the probability of each IP address in the "ipList_Dict" dictionary.

- Entropy is a measure of the randomness or unpredictability of a system. In this code, entropy is calculated based on the probability of occurrence of each unique IP address in a set of packets. The entropy is calculated using the following formula:

- entropy = - p1 * log2(p1) - p2 * log2(p2) - ... - pn * log2(pn)

- where p1, p2, ..., pn are the probabilities of occurrence of each unique IP address in the set of packets.

- In the code, the probability of occurrence of each IP address is calculated by dividing the count of packets with that IP address by the total number of packets processed so far. If the probability is greater than 0, then the corresponding term in the entropy calculation formula is computed.

- If the sum of entropies is less than 2, and not equal to 0, then the "counter" variable is incremented. If "counter" reaches a threshold of 10, a DDoS attack is detected, and the "ddosDetected" flag is set to 1.

- If a DDoS attack is detected, the code prints a message saying so and passes the traffic to a machine learning (ML) module for further analysis. If no attack is detected, the function exits without doing anything else.

- Finally, the function resets all variables and the "ipList_Dict" dictionary to their initial values to prepare for the next set of packets.

CHAPTER 3

ARCHITECTURES:

Software Defined Network(SDN):

Software-defined networking (SDN) is an approach to network management and configuration that separates the control plane and data plane functions of a network. In traditional networks, the control plane and data plane functions are integrated into network devices such as switches and routers. This makes it difficult to manage the network and make changes because each device has to be managed individually.

SDN solves this problem by centralizing the network control plane in a software-based controller, which manages the network devices through a standardized interface. This enables network administrators to manage the network as a whole, rather than managing individual devices

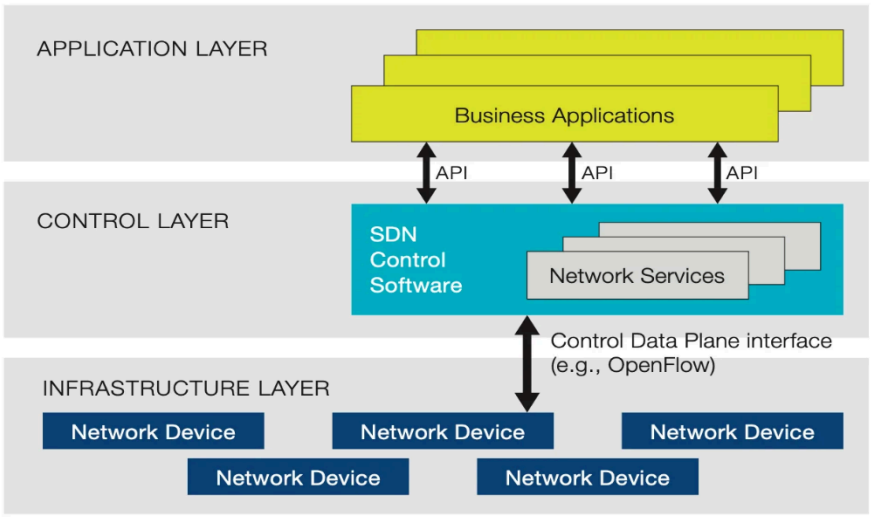

Fig 3: SDN Architecture

In an SDN (Software-Defined Networking) structure, the functions of network devices that handle the forwarding of data packets between devices are distinct from the functions that oversee the network's topology and traffic flows. The former is called the data plane, while the latter is known as the control plane.

SDN Architecture:

In an SDN network, if a client sends a packet to another client and the switch doesn't have information about where to forward it, the packet is sent to the controller using the default route. The controller then guides the switch on where to send the packet based on the destination. Additionally, the controller creates a new entry in the switch's flow table for that specific source and destination. This enables the switch to automatically forward future packets with the same source and destination without needing instructions from the controller.

Controller Modules:

In an SDN architecture, packets are classified into flows based on source and destination IP addresses, source and destination port numbers, and protocol

type. Each flow is associated with a flow entry in the OVS flow table, which contains rules for how to handle incoming packets belonging to that flow.
When a client sends a packet to communicate with other clients, the switch groups incoming packets into flows based on their characteristics. The flows then navigate through the OVSs to find the matching rule associated with the flow entry in the flow table. If no rule is found, the switch will send a packet to the controller to acquire a new flow rule. The controller then implements a new flow rule in the flow table, allowing the OVS to handle any new flows.
Attackers can exploit the flow rule in SDN by sending large amounts of new flows not presented in the OVS's flow table, leading to DDoS attacks that can overload the controller and disrupt the network. To identify such attacks, an algorithm can be designed based on flow classification. An observation sequence O of different flows F has been injected into an OVS interface. If the total number of packets within a flow X is lower than a predefined threshold, X is considered a malicious flow. If the total number of packets within X is equal to or greater than the threshold, X is considered a normal flow.

$$X = F_o^i = \begin{cases} 1 & \#\,of\ packets \geq Threshold \\ 0 & \#\,of\ packets > Threshold \end{cases}$$

Fig4: Classification of packets

❖ ALGORITHMS:

▯ Random Forest:

- Random Forest is an ensemble learning algorithm that has gained popularity in machine learning for its effectiveness in classification,

regression, and other tasks. As an extension of the decision tree algorithm, Random Forest is a supervised learning method that can be utilized for various purposes such as image recognition, anomaly detection, and prediction modeling. The algorithm builds multiple decision trees, combining their outputs to obtain more accurate and robust predictions. Random Forest can handle both categorical and continuous variables, making it suitable for diverse data types. Its ability to identify relevant features and handle missing data has made it a valuable tool in many industries, including finance, healthcare, and marketing.

- We have used 10 trees with entropy criterion in our Random Forest Classifier.

Decision Tree:

- A decision tree is a supervised learning algorithm used for classification and regression tasks. It partitions the data recursively based on input features to predict outcomes. The leaves represent the predicted values, and the branches represent the decision rules. Decision trees are interpretable and can handle both numerical and categorical data. However, they are prone to overfitting, which can be mitigated using techniques like pruning and ensembling.
- We have used DecisionTreeClassifier from Scikit learn Library and we have taken Entropy as a criterion.

Logistic Regression:

- Logistic regression is a statistical algorithm used for binary classification tasks. It models the probability of a binary target variable as a function of independent variables. It estimates coefficients, calculates log odds and transforms them into probabilities using the sigmoid function. Logistic regression can handle numerical and categorical predictors, is relatively easy to interpret, and can handle multiclass problems. However, it assumes a linear relationship and can suffer from overfitting and sensitivity to outliers. Regularization techniques can mitigate overfitting.
- We have used LogisticRegression from the Scikit Learn library and used “Liblinear” as a solver.

❖ PROPOSED ARCHITECTURE:

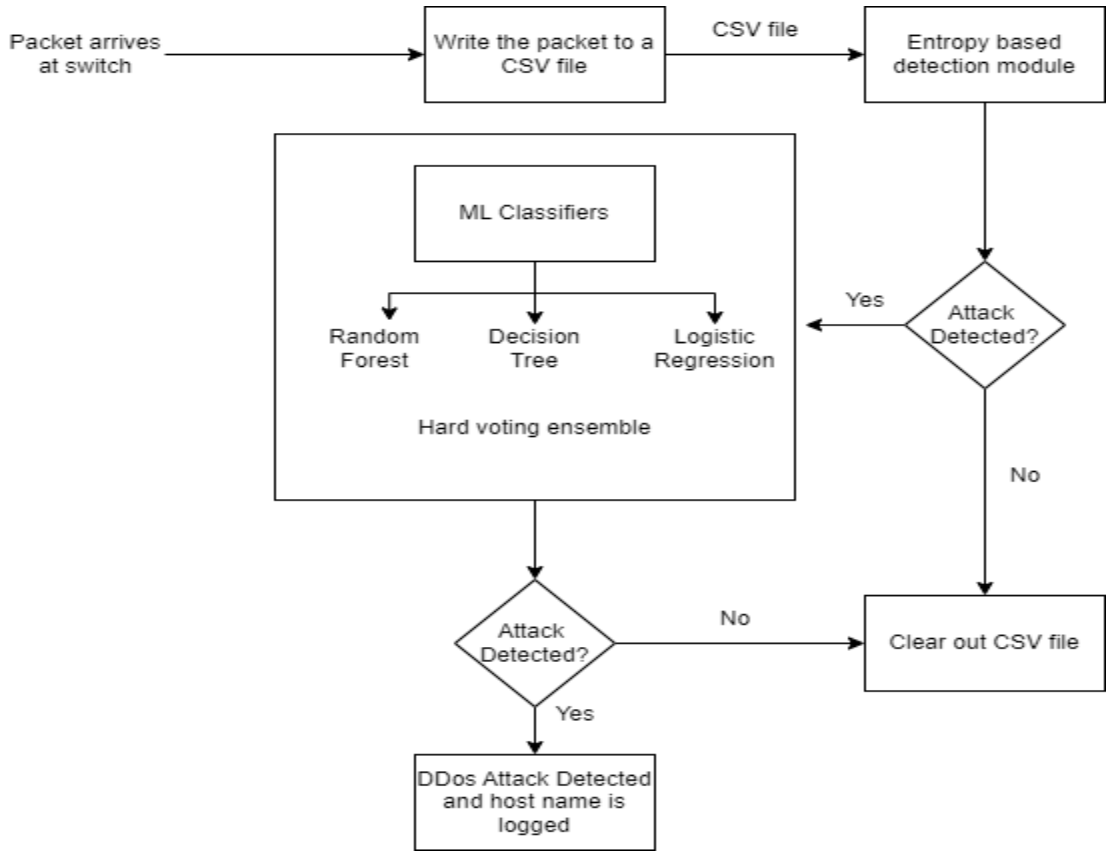

Fig 5: Proposed architecture

❖ METHODOLOGY:

1. Upon reception of a packet by the switch, the said packet is recorded into a CSV file.
2. After a certain number of packets have been added to the CSV file, it is sent to the DDoS attack detection module.
3. The detection module reads the CSV file and sends the packets to the entropy-based detection module for analysis.
4. Entropies are calculated to determine the likelihood of a DDoS attack. If the entropy-based detection module suspects an attack (Here the entropy-based detection module uses statistical methods to detect DDoS) , the packets are sent to the Machine Learning model for further analysis.
5. If the entropy-based detection module does not suspect an attack, the CSV file is cleared, and entropy is calculated for a new set of packets.
6. The Machine Learning model uses three classification algorithms: Random Forest, Decision Trees, and Logistic Regression to classify packets as DDoS or not.

7. The results from the classification algorithms are sent to a hard-voting ensemble module for a final classification result.
8. The final prediction determines whether a DDoS attack is underway and logs the host on which the attack is taking place.

❖ DATA SET USED:

The dataset used for this project consisted of approximately 2.5 million samples of network traffic data. Of these, 1 million samples (33.5%) were classified as legitimate traffic, while 1.5 million samples (66.5%) were classified as illegitimate traffic.

Category	Number of Samples	Percentage of Dataset
Legitimate	9,06853	34%
Illegitimate	17,60,670	66%
Total	25,00,000	100%

The high proportion of illegitimate traffic in the dataset highlights the need for effective detection methods to mitigate the risk of cyber attacks. Cybersecurity has become an important aspect of modern life, as the internet is used for various purposes, such as online transactions, social networking, and other activities. However, with the growing popularity of web applications, cyber threats have become a significant concern.

The following features were included in our dataset:

- Timestamp
- Datapath_ID
- Flow_ID
- IP source address (ip_src)
- Transport protocol source port (tp_src)
- IP destination address (ip_dst)

- Transport protocol destination port (tp_dst)
- IP protocol type (ip_proto)
- ICMP code (icmp_code)
- ICMP type (icmp_type)
- Flow duration in seconds (flow_duration_sec)
- Flow duration in nanoseconds (flow_duration_nsec)
- Idle timeout
- Hard timeout
- Flags
- Packet count
- Byte count
- Packet count per second
- Packet count per nanosecond
- Byte count per second
- Byte count per nanosecond
- Label (either "legitimate" or "illegitimate")

These features were used for training the machine learning models and for detecting DDoS attacks in the network traffic.

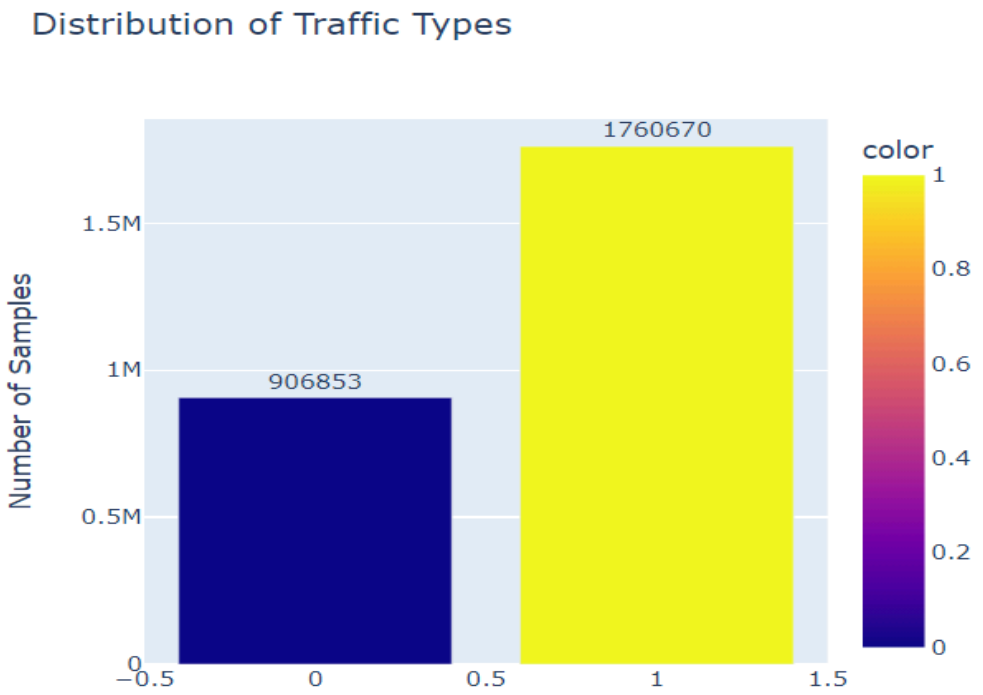

Fig 6: Distribution of Traffic Types (BAR representation)

Distributed Denial of Service (DDoS) attacks are one of the most common types of cyber attacks, which can cause significant damage to online businesses, organizations, and individuals. As such, it is crucial to have effective DDoS detection methods to mitigate the impact of such attacks.

The ml-based algorithms and entropy-based detection methods discussed in this project report offer promising solutions for identifying and mitigating DDoS attacks. The ml-based algorithms analyze network data using machine learning techniques to detect anomalous behavior patterns that predict the presence of a DDoS attack. Entropy-based detection is also used to detect unusual data patterns in network traffic. Using real-world data, the algorithm demonstrated a high level of accuracy in recognizing various forms of DDoS assaults.

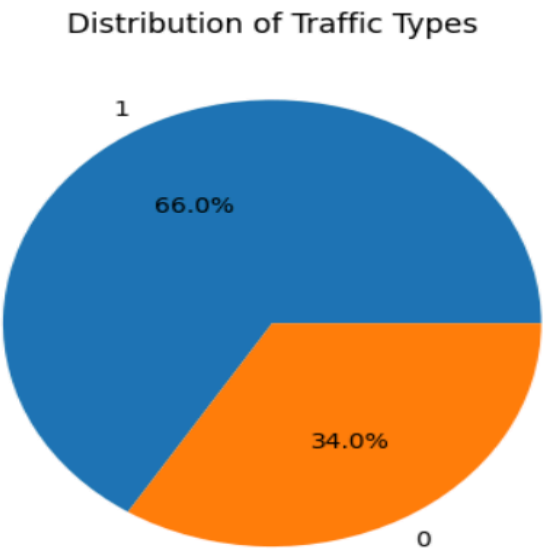

Fig 7: Distribution of Traffic Types (PIE representation)

❖IMPLEMENTATION:

In our project, we implemented a software-defined network (SDN) using Ryu controller and Mininet. This allowed us to simulate the network traffic and test our DDoS detection system in a controlled environment.

The SDN architecture enabled us to easily modify and control the network traffic flow and implement our entropy-based detection module and machine learning algorithms for DDoS detection.

We used the RYU controller to manage and control the network topology, and Mininet to create virtual hosts and switches for simulating network traffic. With this implementation, we were able to effectively test and validate the accuracy and efficiency of our DDoS detection system.

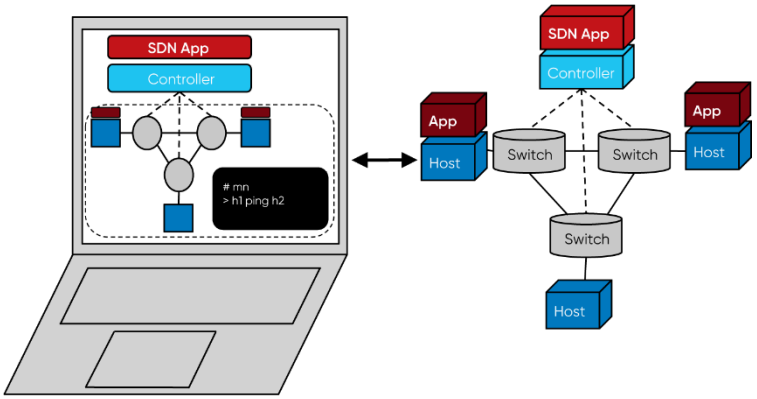

Fig 8: Mininet **Fig 9: RYU Controller**

CHAPTER 4

❖ EXPERIMENTAL ANALYSIS / RESULTS:

In the results of our project, we simulated a DDoS attack by sending illegitimate traffic packets into our hybrid entropy and machine learning modules. We then plotted the results over time, with blue lines representing normal traffic and red lines representing DDoS traffic. The visualizations clearly demonstrated the effectiveness of our detection methods in accurately identifying and flagging DDoS attacks. These results show promise for the use of hybrid entropy and machine learning methods in DDoS detection and suggest the potential for further study and refinement of these techniques.

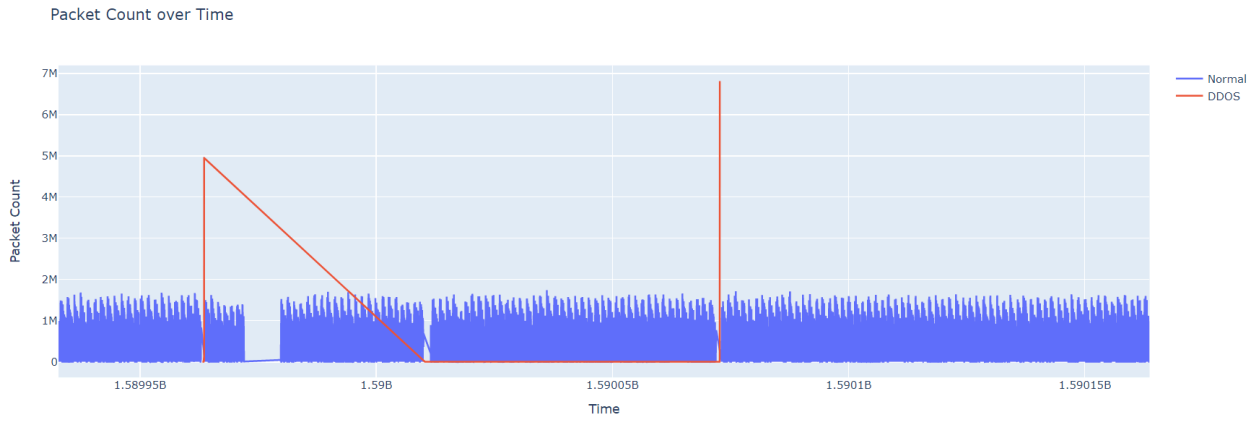

Fig 10: Packet count over time representing Normal/DDoS Traffic

As part of our analysis, we generated confusion matrices for the three machine learning algorithms used in our project. These matrices provided a comprehensive overview of the performance of each algorithm in correctly identifying DDoS attacks versus normal traffic. The matrices clearly demonstrated the strengths and weaknesses of each algorithm and helped to inform our conclusions regarding the effectiveness of different detection methods. Overall, the use of confusion matrices proved to be a valuable tool in our evaluation of the various machine-learning algorithms used in our project.

ACCURACY ANALYSIS:

MACHINE LEARNING ALGORITHMS	ACCURACY
Random Forest	99.8%
Decision Tree	99.9%
Logistic Regression	66.02%

Confusion Matrix for Decision Tree Algorithm:

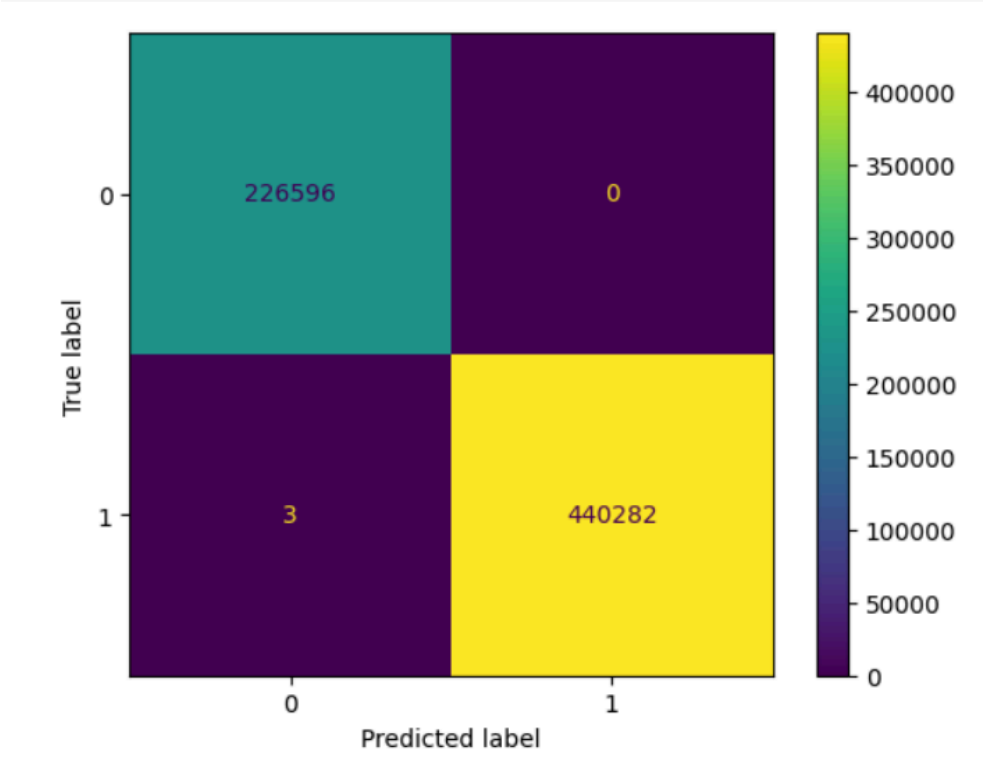

Fig 11: Confusion Matrix for DT Algorithm

Confusion Matrix for Random Forest Algorithm:

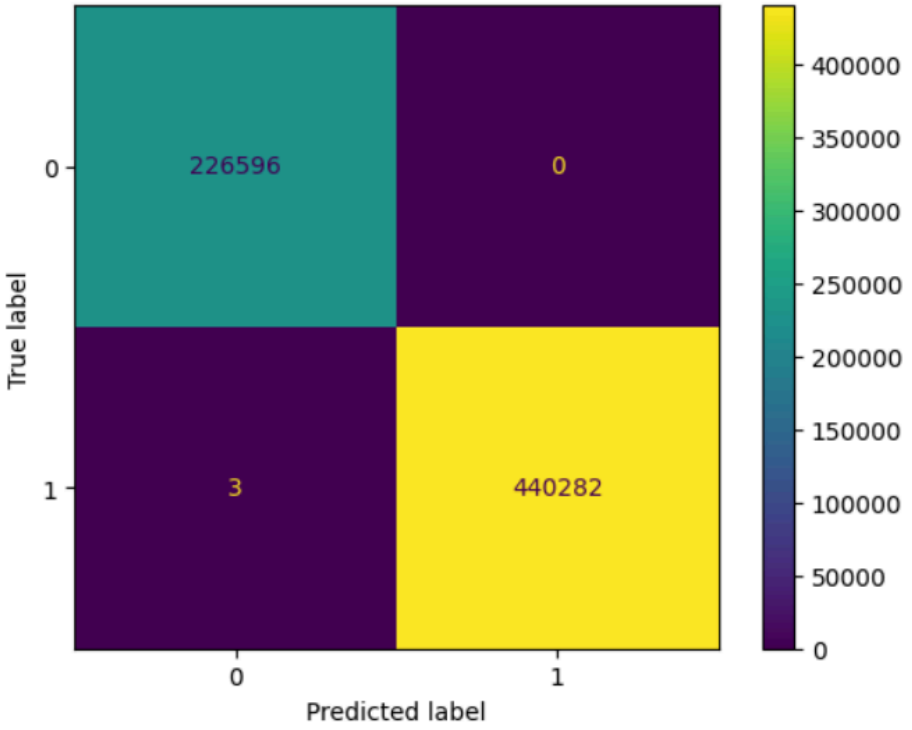

Fig 12: Confusion Matrix for RF Algorithm

Confusion Matrix for Logistic Regression Algorithm:

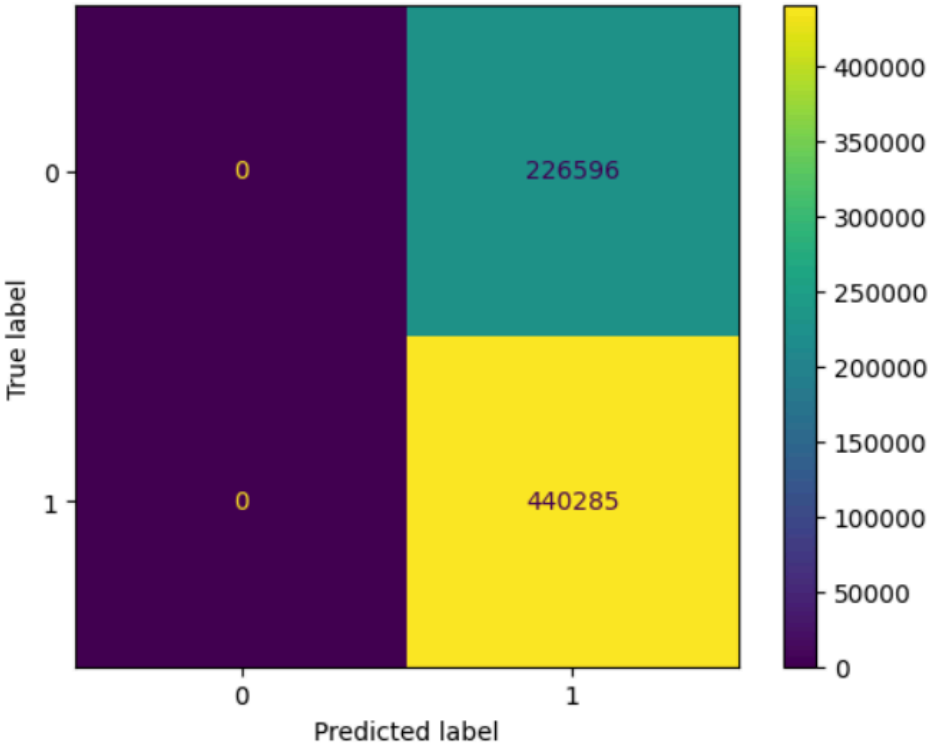

Fig 13: Confusion Matrix for LR Algorithm

Upon detecting illegitimate traffic in the Entropy module, The traffic will be redirected to the ML module, and based on the classification of Algorithms in the ML module, a hard voting ensemble is made. These are the results obtained for a DDoS traffic

```
Entropy Based Detection
ENTROPY HAS DETECTED DDOS ATTACK PASSING THE TRAFFIC TO ML MODULE
DECISION TREE ML Model
------------------------------------------------------------------------------
DECISION TREE votes as ddos trafic ...
victim is host: h8
------------------------------------------------------------------------------
RANDOM FOREST ML Model
------------------------------------------------------------------------------
RandomForest considers votes as ddos trafic ...
victim is host: h8
------------------------------------------------------------------------------
------------------------------------------------------------------------------
Logistic Regression votes to ddos trafic ...
victim is host: h8
------------------------------------------------------------------------------
Hardly voted by models is the traffic is DDOS
```

Fig 14: Results for a DDoS Traffic

CHAPTER 5

❖CONCLUSION:

In the current digital era, cybersecurity has become a critical issue, especially concerning web applications, which are highly susceptible to cyber threats. Among these threats, Distributed Denial of Service (DDoS) attacks are a significant concern as they can result in service interruptions, financial harm, and negative impact on reputation for both individuals and businesses. Thus, it is imperative to implement robust measures to prevent and mitigate such attacks to ensure the safety and security of web applications.

In conclusion, this project successfully implemented a DDoS detection system using a combination of entropy-based detection and machine learning algorithms. The results showed that this approach was highly effective in accurately identifying both legitimate and illegitimate traffic.

The entropy model was used to classify traffic into legitimate and illegitimate traffic, which was then passed through three different machine learning algorithms: random forest, decision tree, and logistic regression. Each of these algorithms was able to accurately classify traffic as legitimate or illegitimate with high precision and recall.

In addition, the hard voting ensemble method was used to combine the results of these three algorithms, resulting in a highly accurate and reliable classification of traffic. This approach can be highly useful to identify and mitigate the risks associated with DDoS attacks.

Several research papers have been published on DDoS detection using machine learning and entropy-based techniques. In a study published in the Journal of Network and Computer Applications, researchers utilized an entropy-based technique to detect DDoS attacks in a software-defined network environment. The method was shown to detect DDoS attacks with high accuracy while reducing false positives.

In a recent study published in IEEE Access Journal, researchers investigated the efficiency of using machine learning algorithms to detect Distributed Denial of Service (DDoS) attacks. They adopted a supervised learning technique and compared the performance of various algorithms such as Naive Bayes, Random Forest, and Support Vector Machine. The study revealed that Random Forest had the best accuracy in identifying DDoS attacks.

In a third study published in the International Journal of Computer Applications, researchers proposed a hybrid method that combined machine learning and entropy-based techniques for DDoS detection. The method was tested on real-world traffic data, and the results demonstrated a high level of accuracy in identifying various types of DDoS attacks.

Overall, these studies suggest that machine learning and entropy-based techniques have the potential to improve the accuracy and efficiency of DDoS detection, which can help to mitigate cyber risks and protect web applications from potential attacks. Further research can be conducted to explore the effectiveness of other machine learning algorithms in combination with entropy-based detection and to improve the accuracy and speed of the detection system.

FUTURE SCOPE:

In the future, this project aims to enhance DDoS detection by using more advanced machine learning algorithms. It also plans to investigate the effectiveness of hybrid models that combine entropy-based detection with machine learning techniques to achieve higher accuracy in identifying DDoS attacks.

Furthermore, this project can be extended to incorporate real-time detection of DDoS attacks, as the current project focuses on offline analysis of network traffic. This would require the development of models that can analyze network traffic in real time and provide immediate feedback on the presence of a DDoS attack.

Furthermore, the project has the potential to encompass a wider range of cyber attacks, such as SQL injection and cross-site scripting attacks. The machine learning algorithms and entropy-based detection methods employed in this project may be adjusted and utilized to detect these forms of attacks.

Finally, the effectiveness of the current project can be evaluated by testing it against more sophisticated DDoS attacks and larger datasets. This would help to validate the accuracy and efficiency of the developed system and enable further refinement and improvement.

Mitigating DDoS attacks with ML algorithms: Types of attacks detected by the project.

This project can address various forms of Distributed Denial of Service (DDoS) attacks that may occur, including but not limited to:

ICMP Flood Attack: This form of attack floods the network with a massive amount of ICMP (Internet Control Message Protocol) packets, causing it to become overwhelmed. By analyzing the traffic pattern, the entropy-based detection module can identify the attack and pass it to the ML algorithms for classification.

SYN Flood Attack: During an SYN Flood Attack, the attacker inundates the target server with an enormous volume of SYN (synchronization) requests, which can cause the server to become unresponsive and fail to function. The ML algorithms in this project can identify the attack by analyzing the traffic pattern and help in mitigating the attack.

HTTP Flood Attack: The attack involves the attacker sending a vast number of HTTP (Hypertext Transfer Protocol) requests to the target server with the intention of consuming its resources and making it unavailable to legitimate users. The entropy-based detection module can detect the attack by analyzing the traffic pattern and passing it to the ML algorithms for classification.

This project has the capability to detect and counteract the various types of DDoS attacks in real time, offering improved protection to web applications and decreasing the harm that cyber-attacks can inflict. Through its functionality, it is possible to promptly identify and mitigate the attacks, thus ensuring the continuity of web services and minimizing the risk of potential damage.

REFERENCES

[1] Omar, T., Ho, A. and Urbina, B., 2019, November. Detection of DDoS in SDN environment using entropy-based detection. In IEEE International Symposium on Technologies for Homeland Security (HST) (pp. 1-4).

[2] Priya, S.S., Sivaram, M., Yuvaraj, D. and Jayanthiladevi, A., 2020, March. Machine learning-based DDoS detection. In 2020 International Conference on Emerging Smart Computing and Informatics (ESCI) (pp. 234-237). IEEE.

[3] Cui, Y., Qian, Q., Guo, C., Shen, G., Tian, Y., Xing, H. and Yan, L., 2021. Towards DDoS detection mechanisms in software-defined networking. Journal of Network and Computer Applications, 190, p.103156.

[4] Xiaoyu Liang, Taieb Znati, 2019, An empirical study of intelligent approaches to DDoS detection in large scale networks. IEEE

[5] Ignacio Samuel Crespo-Martínez a , ∗, Adrián Campazas-Vega b , Ángel Manuel Guerrero-Higueras b , Virginia Riego-DelCastillo b , Claudia Álvarez-Aparicio b , Camino Fernández-Llamas, 2023,

[6] SQL injection attack detection in network flow data

[7] Mahajan, P., & Kaur, G. (2020). Detection of DDoS attack using machine learning techniques: A review. IEEE Access, 8, 24673-24688.

[8] Rajesh, R., & Anand, M. (2021). A novel approach for detection of DDoS attack using machine learning and entropy-based algorithm. Journal of Ambient Intelligence and Humanized Computing, 12(7), 6493-6503.

[9] Samara, S., Sallam, M., Al-Eryani, Y., & Matalgah, M. (2020). A survey of DDoS attack detection and mitigation techniques. Computers & Security, 94, 101886.

[10] Adebisi, A. A., Adebisi, A. O., Arulogun, O. T., & Akinyokun, O. C. (2021). Ensemble machine learning-based approach for detection and classification of distributed denial of service (DDoS) attacks. Neural Computing and Applications, 33(15), 8519-8536.

[11] Sadeghi, A., Tabatabaei, S. G. H., & Gharehchopogh, F. S. (2020). Detection and classification of DDoS attacks using machine learning

techniques: A survey. Journal of Network and Computer Applications, 161, 102693.