CSE 543: Information Assurance and Security

Group 1 - 5: Evaluating Machine Learning-based approaches that detect DDoS Attacks in Cloud Systems

List of members:

ASU ID	Name	Email Address	Role
1225625856	Harshit Viren Shah	hshah78@asu.edu	Leader
1224141347	Mahesh Chandra Yayi	myayi@asu.edu	Deputy Leader
1222228566	Manoj Boddu	mboddu1@asu.edu	Member
1211219734	Prakruthi Ravandur Madesh	pravandu@asu.edu	Member
1225329573	Rama Narasimhan Magesh	rmagesh@asu.edu	Member
1212760729	Ricardo Lizarraga	lizar1@asu.edu	Member
1227953352	Asutosh Karanam	akarana5@asu.edu	Member
1222128713	Pruthvi Patel	pjpate12@asu.edu	Member

Table of Contents:

1. Introduction **4**

1.1. Motivation 4

1.2. Scope of Study 5

2. Summary of Accomplishments of the Project **6**

3. Accomplishments of each group member **7**

4. Detailed Results of all individual group members **10**

4.1. Types of DDoS Attacks **10**

4.2. Classification on Protocol level **10**

4.2.1. Network/transport-level DDoS flooding attacks 10

4.2.2. Application-level DDoS attacks 12

4.3. Traditional DDoS Detection techniques **13**

4.3.1. Firewall 13

4.3.2. IDS and IPS 15

4.3.3. Network Security Groups 17

4.3.4. Why do they fail in cloud computing? 18

4.4. Detecting DDoS Attacks Using Machine Learning **18**

4.4.1. Detection from Cloud Source 19

4.3.2.1 Detecting different DDoS Attacks from the Source Side 20

4.3.2.1 Algorithms used to detect from Source side 23

4.4.2. Cloud as a Destination 24

4.5. Problems with Machine Learning for DDoS Detection **27**

4.6. Optimizing Machine Learning Algorithms **28**

4.6.1. Forward Feature Selection 28

4.6.2. Supervised Machine Learning in conjunction with ANN 29

4.6.3. Hybridized Machine learning IDS models 32

4.6.4. Perplexed Bayes Classifier 34

4.7. Deep Learning for DDoS Detection **36**

4.7.1. Detecting DDoS Attacks at Gateway Router using Deep Learning 37

4.7.2. Detecting DDoS Attacks with a Hybrid Ensemble Model 38
4.7.3. DDoS Attack Detection Method Based on Deep Transfer Learning 40
4.7.4. Detecting DDoS Attacks using FedAvg and FLAD 41
4.7.5. Detecting DDoS Attacks with Hierarchical Temporal Memory 44
4.8. Challenges for future research 45
4.9. Results 46
5. Conclusions and Recommendations 48
6. References 51

1. Introduction

The emergence of cloud computing has revolutionized the way applications are used by users and built by businesses. It has enabled organizations to manage large projects remotely and stay connected with their clients. Despite the numerous benefits, there are malicious individuals who seek to exploit cloud computing technology for their own gain. One of the most dangerous ways they achieve this is through Distributed Denial of Service (DDoS) attacks.

DDoS attacks are a common type of cyber attack that can cause severe financial losses and reputational damage to a company. These attacks overwhelm servers or networks with a high volume of requests, rendering them unavailable to legitimate users. DDoS attacks can be particularly challenging to prevent as they often involve multiple machines. In cloud computing systems, DDoS attacks can be even more devastating as they can use multiple connections to inflict harm on businesses relying on cloud platforms.

To combat these malicious attacks, machine learning algorithms can be used to enhance cloud computing security. Machine learning continues to make significant strides in recent years, and with the application of such algorithms, it is possible to improve current security measures. The report proposes using supervised machine learning algorithms to detect DDoS attacks. It discusses the inadequacies of traditional methods that are currently used by cloud platforms to prevent DDoS attacks and explores the alternative approach that uses machine learning algorithms that can be used to detect these attacks effectively.

1.1. Motivation

The current defenses used in enterprise network security are inadequate against the increasing complexity and sophistication of DDoS attacks. Therefore, more effective defenses are required, such as machine learning algorithms that excel in recognizing patterns. These algorithms have proven to be successful in predicting accurate results in different fields, given sufficient data. Hence, they can play a vital role in building more secure defenses against DDoS attacks.

Understanding the different types of DDoS attacks on cloud platforms and evaluating the efficiency of machine learning algorithms in detecting such attacks can ensure the reliability, availability, and security of cloud computing platforms. To reiterate, DDoS attacks can disrupt cloud services, leading to significant losses in revenue, reputational damage, and even legal liabilities. By analyzing network traffic patterns and identifying abnormal behavior, machine learning algorithms can detect and mitigate such attacks. Assessing the effectiveness of machine learning algorithms in detecting DDoS attacks can help develop

more robust and accurate detection systems that can proactively respond to attacks, ensuring the continuity of cloud services. This is particularly important in the current digital landscape, where cloud platforms are increasingly critical for business operations.

The team's main objective is to conduct a literature review of various types of DDoS attacks that could compromise a cloud computing platform and assess the effectiveness of machine learning-based approaches in detecting and mitigating such attacks. This study will take the initial steps towards developing more resilient cybersecurity defenses to protect cloud platforms from the damaging effects of DDoS attacks.

1.2. Scope of Study

- In this study, we primarily focus on DDoS attacks as they have debilitating effects in terms of revenue as well as the difficulty involved in detecting and mitigating compared to DoS attacks.
- Survey of cloud platforms' vulnerability to various types of DDoS Attacks such as UDP floods, ICMP floods, SYN floods, fragmented packet attacks, Ping of Death, Smurf DDoS, low-and-slow attacks, GET/POST floods and the current state-of-the-art security measures they employ that are not machine learning-based with a goal to evaluate their effectiveness.
- Survey the traditional forms of defense against such attacks and identify their shortcomings.
- Identify how machine learning fits into the realm of security in regards to these various types of DDoS attacks.
- Investigate various machine learning-based approaches for detecting and mitigating DDoS attacks for both source and destination sides.
- Explore how deep learning can be used in DDoS attack detection and where it can be applied.
- Summarize findings, create a comprehensive list of machine learning algorithms that are well-adapted to detect DDoS attacks, and compare the performance of each ML algorithm against diverse types of DDoS attacks.
- Assessment of future trajectory of security measures, in terms of the technologies that can be leveraged, which will be employed by the cloud platforms.

2. Summary of Accomplishments of the Project

- Condensed numerous research papers into a cohesive report based around the subject of info assurance.
- Identified the different types of DDoS attacks, how they can be used to attack cloud systems by conducting a literature review on the taxonomy of DDoS attacks.
- Surveyed various classification schema of DDoS attacks based on attack source, attack vector, and attack target.
- Came to the conclusion, as a team, that classifying the DDoS attacks based on attack vector is the most comprehensive way to classify all DDoS attacks and surveyed the traditional defense mechanisms that cloud computing platforms currently employ for detecting such attacks.
- Categorized the defense mechanisms based on the same classification schema used for DDoS attack taxonomy, and assessed the efficacy of each of them.
- Detecting DDoS attacks on both the destination and source sides of cloud platforms is crucial, and traditional methods of detecting attacks around cloud servers may not be sufficient.
- Came to the conclusion, as a team, that the traditional approaches for detecting DDoS attacks are not adequate and identified machine learning-based algorithms as an alternative approach to defend cloud platforms against DDoS attacks.
- Conducted literature survey on various types of machine learning algorithms that would be suitable for predicting malicious network patterns that would thwart oncoming DDoS attacks.
- Categorized the suitable machine learning algorithms from domains such as clustering, deep learning, etc. into Supervised and Unsupervised learning algorithms.
- Conducted a literature review on the effectiveness of these algorithms to detect and mitigate DDoS attacks for a comparative evaluation of their performances.
- Presented relevant information like datasets, data preprocessing methodologies, and ML model architecture.
- Learned the strengths and weaknesses of supervised machine learning and unsupervised machine learning in regards to DDoS attack detection.
- Surveyed literature on comparative evaluation of performances between traditional machine learning algorithms and certain hybrid machine learning models to identify stronger defense measures against DDoS attacks and explore the potential to further improve DDoS detection.
- Came to the conclusion that machine learning algorithms are essential for DDoS attack detection and discussed future directions for research, including novel ML models and technologies to build better intelligent defense systems.

3. Accomplishments of each group member

Harshit Viren Shah:

- Identified the research papers for the topic reviewing the initial information and assigning papers to team members for in depth study
- Completed a literature review for various topics identified by the team - traditional cloud DDoS protection strategies, source side DDoS detection using Machine Learning, using deep learning for DDoS detection in cloud systems, challenges for supervised machine learning.
- Schedule regular meetings to monitor progress of the project.
- Scheduling the tasks for the project, managing deadlines and the overall project progress.
- Scrutinize the work among team members approving the study reports. Evaluated and approved in-depth study reports created by the team members.

Mahesh Chandra Yayi

- Coordinating with the team leader to ensure that the project is on track and meeting its objectives.
- Facilitating communication between team members to ensure that everyone is aware of their roles and responsibilities.
- Assisting with the development and implementation of project plans and timelines by participating in team meetings and discussions.
- Discussed and followed up on various team responsibilities with the project leader.
- Completed a comprehensive literature review that introduces, classifies a plethora of DDoS attacks and presents the cutting edge research on machine learning-based algorithms that detect and mitigate various kinds of sophisticated DDoS attacks on cloud computing infrastructure.
- Performed numerous casual and in-depth reviews of papers that focus on classifying DDoS attacks and automated detection using machine learning-based algorithms.
- Evaluated and approved several in-depth reports of papers that we chose as a team for this project.

Ricardo Lizarraga:

- Performed in-depth studies on DDoS attacks on the source side, including how to detect and prevent them from harming victims.
- Researched how different machine-learning algorithms can be utilized on different types of attacks within a cloud server and how the different types of algorithms could be employed to detect these attacks.
- Assisted in approvals where the team leader could not review their own work.

- Participated in all group discussions and class sessions, sharing information to those that could not attend.

Asutosh Karanam

- Identified relevant research papers, read them roughly, and extracted relevant information to determine if they would help in our research activity.
- Visited the Writing center and sought to acquire knowledge on report creation. This involved understanding best practices in research, attending tutoring sessions and collaborating with subject matter experts.
- Conducted literature reviews, attended online webinars, and collaborated with other researchers. Gained a better understanding of the state-of-the-art Machine Learning techniques currently in vogue and how hybridizing ML models perform in the realm of Intrusion detection.
- Studied on the available tools that simulate the attack to evaluate the proposed ML models and cross-check to confirm the results.
- Actively participated in all of the Group Discussions and contributed to multiple tasks of pitching the project idea, ideating the extensive scope, report creation etc.

Pruthvi Patel

- Collected and identified papers relevant to the project title. Conducted in-depth studies on traditional detection techniques for DDoS attacks.
- Evaluated in-depth studies of other group members and also made sure that the evaluations follow the guidelines set out.
- Taken responsibility for authoring a weekly report and coordinated with team members to make sure it is in line with guidelines and what they contributed that week.
- Contributed to group discussions both in-person and online to guide those discussions towards the aim of the project.

Manoj Boddu

- Conducted comprehensive analysis of other group members in-depth studies, ensuring that they adhered to the established guidelines.
- Played an active role in all weekly group discussions and contributed to various tasks.
- Collected different papers related to DDoS detection which can be useful to the project
- Performed in-depth studies on security challenges in cloud computing and adaptive federated learning for DDoS attack detection.

Prakruthi Ravandur Madesh

- Identified pertinent research papers, skimmed them, and extracted pertinent data to see if they would be useful for our study activity.
- Conducted in-depth research on DDoS assaults on cloud systems, including how to recognize them and stop victims from suffering harm.
- Reviewed the literature and worked with other researchers, improved knowledge of the current cutting-edge neural network and deep learning intrusion detection methods.
- Reviewed various publications that deal with identifying DDoS attacks and automating detection using machine learning-based methods, both casually and in-depth.
- Evaluated a few in-depth analyses of the publications that our team had selected for this project.
- Participated actively in all weekly group talks and helped with various kinds of tasks in projects.

Rama Narasimhan Magesh

- Performed a high-level analysis on the research papers and determined if they would be useful for the given research topic.
- Performed in-depth studies on using various Machine learning methods to identify DDoS attacks on source side, including C4.5 classifier and Perplexed Bayes classifier algorithms, understood the underlying concepts presented in the papers and wrote reports on the same.
- Evaluated in-depth studies of other group members and also made sure that the evaluations follow the guidelines set out.
- Took up the task of creating a weekly report and synced up with teammates regarding the progress made during the given week.
- Actively participated in group discussions, both virtual and in-person, and provided updates to the team members regarding the progress made, and asked any questions if needed.

4. Detailed Results of all individual group members

4.1. Types of DDoS Attacks

The taxonomy of DDoS attacks is classified based on the attack source, attack vector, and the attack target. Attack sources can be classified as botnets, mobile devices, and Internet of Things (IoT) devices. Attack vectors can be classified as protocol-based, and application-based. Attack targets can be classified as network infrastructure, web servers, and application servers. Survey papers that focus on classifying DDoS attacks based on attack vectors are prioritized for literature review as this allows for a broader and more hierarchical classification. Both in-depth and brief reviews of many reputed academic research papers on DDoS classification, such as [23][24][25][27][28], are completed and understand that it is possible to pool and aggregate the various types of DDoS attacks. The classification schema of DDoS attacks selected for this report is from Zargar et al.[25] as it is broader in scope, more robust and future-proof compared to the ones presented in Tasnuva et al.[23] and Christos et al.[24] Figure 1 is a hierarchical tree representation of DDoS attack classifications presented in Zargar et al.. DDoS attacks are primarily classified into 2 attack vector categories, which are network layer attacks and application layer attacks.

4.2. Classification on Protocol level

The two groups of DDoS flooding attacks based on the protocol level that is targeted are:

- Network/transport-level DDoS flooding attacks: These attacks flood a network or transport layer protocol with traffic. Examples include TCP SYN floods, UDP floods, ICMP floods, and DNS amplification attacks.
- Application-level DDoS attacks: These attacks target the application layer of a server or web application. They are designed to consume server resources, such as CPU or memory, rather than bandwidth. Examples include HTTP floods, Slowloris attacks, and DNS query attacks.

4.2.1. Network/transport-level DDoS flooding attacks

DDoS flooding attacks at the network/transport-level are typically executed through the use of TCP, UDP, ICMP, and DNS protocol packets. This category encompasses four distinct types of attacks which are presented in detail in [24][29][30][31].

In terms of DDoS flooding attacks, there are four main categories:

Flooding attacks: Attackers focus on disrupting legitimate user's connectivity by exhausting the victim network's bandwidth. Examples include Spoofed/non-spoofed UDP flood, ICMP flood, DNS flood, VoIP Flood, etc.

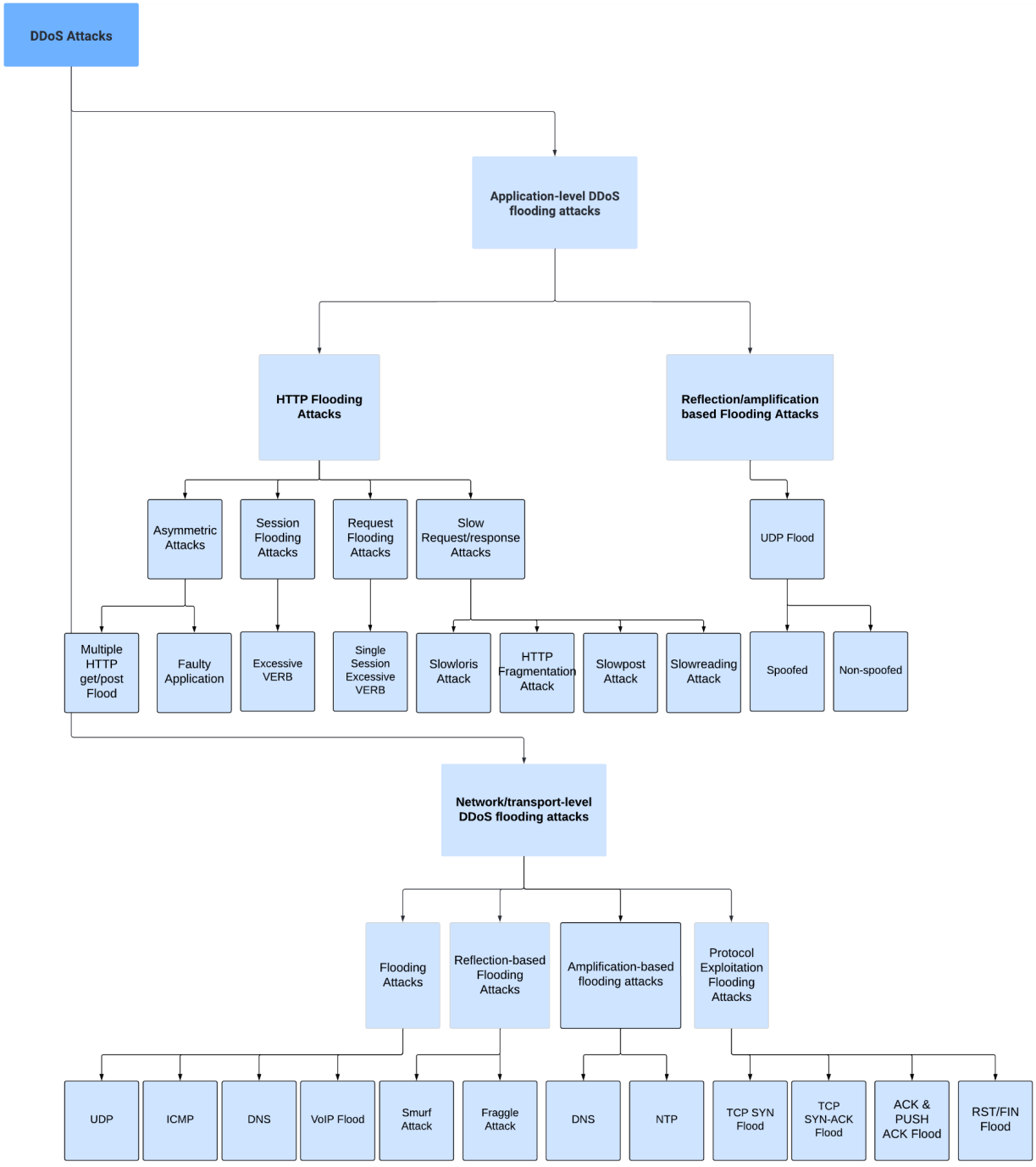

Fig. 1: *DDoS attack classification hierarchy tree.*

Attackers use different techniques to flood a victim's resources. In Protocol Exploitation Flooding attacks, attackers take advantage of implementation bugs or specific features of the victim's protocols to consume excessive amounts of resources. These attacks include TCP SYN flood, TCP SYN-ACK flood, ACK & PUSH ACK flood, RST/FIN flood, among others.

Reflection-Based Flooding attacks involve attackers sending forged requests to reflectors, causing them to send replies to the victim and exhaust the victim's resources. Smurf and Fraggle attacks are examples of Reflection-Based Flooding attacks.

In Amplification-Based Flooding attacks, attackers exploit services to generate large or multiple messages for each message they receive, amplifying the traffic towards the victim. Examples of Amplification-Based Flooding attacks include DNS amplification, NTP amplification, etc.

Reflection and Amplification techniques are often used in tandem in attacks such as Smurf attack. In these attacks, attackers use spoofed source IP addresses (reflection) to send requests to a large number of reflectors, exploiting the IP broadcast features of the packets (amplification).

4.2.2. Application-level DDoS attacks

Application-level DDoS attacks target server resources in order to disrupt legitimate user services [32]. These attacks target specific characteristics of applications, such as HTTP, DNS, or Session Initiation Protocol (SIP), and are stealthier in nature compared to volumetric attacks. However, they usually have the same impact on services. One example of an application-level flooding attack is the DNS amplification flooding attack and the SIP flooding attack, which are famous reflection/amplification flooding attacks targeting DNS and SIP protocols. Another commonly reported type of DDoS flooding attack involves the HTTP protocol which is further classified into four subtypes [33].

1. **Reflection/amplification based flooding attacks:**

These attacks use similar methods to those used at the network and transport levels, as presented in [24] and [29], involving falsified requests for application-level protocols to overwhelm a target system. Examples include DNS amplification and VoIP flooding, which use reflection techniques to generate a large volume of traffic with fake IP addresses. These attacks can be difficult to identify as they mimic legitimate traffic.

2. **HTTP flooding attacks:**

These attacks can further be classified into the following types [31].

- **Session flooding attacks:** Attackers overwhelm a server by requesting sessions at a higher rate than legitimate users, causing a DDoS flooding attack. An example is the HTTP get/post flooding attack, where attackers send many valid HTTP requests to a web server using botnets. Only a few bots are needed for a successful attack, and these attacks are not spoofed.
- **Request flooding attacks:** Attackers use this attack to flood a server by sending more requests per session than usual. One example is the single-session HTTP get/post flooding attack, a variation of the HTTP get/post flooding attack that takes advantage of the multiple request feature in HTTP 1.1. By limiting the session rate, attackers can bypass defense mechanisms that rely on session rate limitation.
- **Asymmetric attacks:** In this attack, high-workload requests are sent by the attackers. Two well-known attacks in this category are the Multiple HTTP get/post flood attack and the Faulty Application attack. In the Multiple HTTP get/post flood attack, the attacker sends multiple HTTP requests in a single packet, which maintains high loads on the victim server with a low attack packet rate, making it difficult to detect. In the Faulty Application attack, attackers take advantage of poorly designed or integrated websites to lock up database queries using SQL-like injections, consuming server resources.
- **Slow request/response attacks:** Attackers can send high-workload requests in this type of attack. Famous attacks in this category include:
    - **Slowloris attack:** HTTP get-based attack that sends partial HTTP requests continuously and rapidly grows, causing the Web server to become inaccessible.
    - **HTTP fragmentation attack:** attack where attackers establish a valid HTTP connection with a Web server and fragment legitimate HTTP packets into tiny fragments, bringing down the server with just a handful of bots.
    - **Slowpost attack:** sends HTTP post commands slowly, causing the server to wait for each message body to be completed while the attack grows rapidly.
    - **Slowreading attack:** works by slowly reading the response instead of slowly sending the requests, forcing the server to keep a large number of connections open and eventually causing it to crash.

There are many defenses against these attacks and the next section gives an overview of traditional DDoS detection techniques.

4.3. Traditional DDoS Detection techniques

4.3.1. Firewall

A firewall is a type of network security system that keeps track of and manages incoming and outgoing network traffic in accordance with a predetermined set of security rules. It acts as a barrier between a private network and the public internet by preventing unauthorized access and protecting the network from outside threats. By utilizing several strategies including rate-limiting, traffic filtering, and traffic shaping, firewalls can stop DDoS attacks. Limiting the amount of traffic that is permitted to enter the network by rate-limiting can assist keep the network from being overloaded. Blocking traffic that is connected with DDoS assaults, such as packets with fake IP addresses, is known as traffic filtering. By giving some forms of traffic precedence over others, traffic shaping can assist to reduce congestion and guarantee that legal traffic is given priority. In order to offer more thorough protection against DDoS assaults, firewalls can also be used in concert with other network security solutions like load balancers and web application firewalls. By distributing traffic among several servers, load balancers can lessen the risk of a DDoS assault overwhelming a single server.

Signature-based detection

By comparing information to a database of recognized signatures or patterns of malicious behavior, firewalls may identify malicious traffic using the signature-based detection technique. The database includes details on the characteristics of many attack types, including malware, viruses, and worms. The firewall checks the incoming traffic for matches with the database when it gets it. The firewall can block the traffic, notify the administrator, or take any necessary action if a match is discovered. Efficacy of signature-based detection in identifying and thwarting DDoS assaults: Signature-based detection is very successful in thwarting DDoS attacks. DDoS attacks usually entail saturating the target network with a lot of traffic in order to block access for users. By comparing it to recognized patterns of DDoS assaults, signature-based detection may recognize and stop this traffic. Since the database's signatures are distinct to each sort of attack, signature-based detection is very effective at recognizing known threats, such as DDoS attacks. Also, the database contains predetermined signatures, the firewall can instantly match incoming traffic against the database and take fast action, reducing the harm caused by DDoS assaults.

Threshold-based detection

A threshold-based detection is a popular approach in signal processing and machine learning for detecting the presence of a certain signal or event in a noisy signal. It entails establishing a threshold level beyond which a signal is deemed present and below which it is deemed missing. To improve detection performance, the threshold can be adjusted based on past information or statistical methodologies.

Threshold-based detection requires comparing a signal's amplitude to a defined threshold level. If the amplitude is greater than the threshold, the signal is regarded as present; otherwise, the signal is considered missing. This method is employed in a variety of applications, including communication system signal detection, image processing, biological signal analysis, and voice recognition.

In simpler terms, the threshold-based detection technique works like this:

- First, choose a threshold level.
- Then, measure the signal's amplitude.
- Next, compare the amplitude to the threshold level.
- If the amplitude is higher than the threshold, the signal is present. If it's lower, then the signal is absent.

The simplicity of threshold-based detection is one of its key advantages. The approach is simple to construct and may be used in a number of circumstances. It also does not need any sophisticated signal processing techniques or machine learning models, making it a low-cost option. The speed of threshold-based detection is another advantage. The approach is useful for real-time applications such as signal detection in communication systems because it can swiftly determine the existence or absence of a signal.

Anomaly detection

Anomaly detection is a method for detecting odd behavior patterns in a system or network that may signal the presence of a security problem. In the context of firewall security, anomaly detection is a valuable tool for recognizing Distributed Denial of Service (DDoS) assaults. Anomaly detection techniques can aid in the identification and mitigation of distributed denial of service attacks, hence increasing network security. Anomaly detection techniques are extensively employed in DDoS detection for firewall protection, and they include traffic analysis which is the process of examining network data to find trends that may indicate a DDoS assault. A rapid increase in traffic from a single IP address, for example, may indicate the presence of a botnet. Packet inspection entails inspecting the contents of individual packets to find patterns that may signal a DDoS assault. Packets carrying a significant amount of identical requests, for example, may signal the presence of a DDoS assault. Behavioral analysis is the process of monitoring network activity in order to find patterns that indicate a security concern. For example, if a network device suddenly starts transmitting significant volumes of traffic to an unknown location, this might indicate that the device has been hacked. Anomaly detection systems can detect DDoS assaults early in the process, allowing network administrators to take immediate action to neutralize the attack. Anomaly detection techniques are intended to limit false positives by filtering out regular traffic patterns and only

detecting aberrant activity. Increased Accuracy: Anomaly detection techniques can enhance DDoS detection accuracy by spotting patterns that typical security solutions may overlook.

4.3.2. IDS and IPS

An Intrusion Detection Systems (IDS) is a security tool that monitors network traffic for suspicious activity, such as unusual login attempts or data transfers, and alerts security administrators when it detects such activity. IDS can be host-based, which means it monitors individual devices, or network-based, which means it monitors the entire network. IDS can be classified into different categories, such as signature-based, anomaly-based, or hybrid systems.

An Intrusion Prevention Systems (IPS), on the other hand, not only detects suspicious activity but also takes action to prevent it from succeeding. IPS monitors network traffic and can automatically block traffic from known malicious IP addresses or patterns of traffic that suggest an attack. IPS can also be host-based or network-based, and can use various techniques such as stateful inspection, deep packet inspection, or behavior-based detection to identify and prevent attacks.

As Jema David Ndibwile et.al established in [21], Several mitigation strategies of using IDS in conjunction with a tri-server architecture are popular wherein the situation involves the communication between a client's or an attacker's computers through a Bait Web server, which is publicly visible unlike the Decoy and Real Web servers. The Real Web server is set up to host a fully functional website and receive all incoming requests via the Bait Web server. Meanwhile, the Bait Web server is configured to mainly listen for incoming requests on the standard port from the external world and operate as a proxy. Bait Web server determines what to do with the received traffic; in this case it sends authenticated traffic to the Real Web server and unauthenticated traffic to the Decoy Web server, depending on the user's action on the authentication method. The datapath of the traffic is as shown in Fig. 2 below.

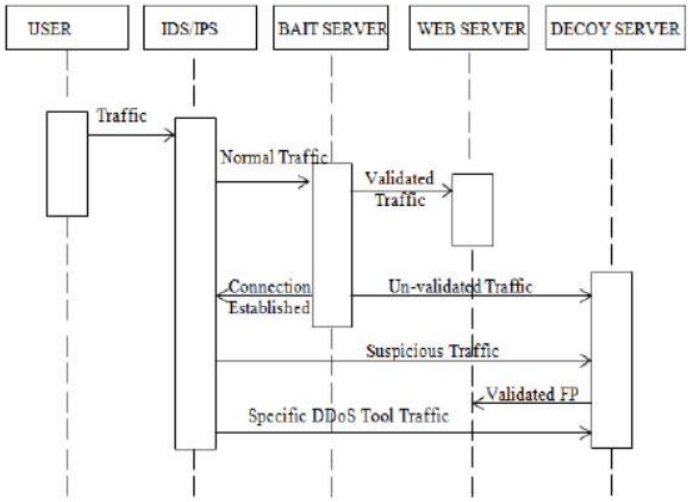

Fig. 2: *Commonly used Snort Intrusion Detection System - Tri level Architecture.*

The use of machine learning in IDS and IPS has emerged as a promising approach to improving the accuracy and effectiveness of these systems. Machine learning algorithms can analyze large volumes of data and identify patterns that may indicate an attack, even if the attack is previously unknown. By continuously learning and adapting to new data, machine learning models can improve their accuracy over time and keep pace with evolving cyber threats. Putting ML algorithms into use by the IDS systems has been an evolving field of Research.

4.3.3. Network Security Groups

Network Security Groups (NSGs) on Azure virtual machines may be used to identify and mitigate Distributed Denial of Service (DDoS) attacks. Here are some approaches for detecting DDoS assaults that may be utilized with NSGs:

- **Rate limiting:** NSGs can be set to limit the amount of traffic that can travel into and out of a virtual network. By restricting the amount of bandwidth that may be transmitted to a virtual machine, this can assist avoid DDoS assaults.
- **Whitelisting:** NSGs can be set to accept only traffic from known sources, such as specified IP addresses or ranges. By limiting communication from known malicious sources, this can assist avoid DDoS assaults.
- **Blacklisting:** NSGs can also be configured to prevent traffic from known malicious sources via blacklisting. This can aid in the prevention of DDoS attacks by limiting bandwidth from known DDoS attack sources.

- **Protocol filtering:** NSGs can be set to accept or reject certain protocols like TCP or UDP. This can aid in the prevention of DDoS assaults, which employ specialized protocols to overwhelm a virtual machine with traffic.
- **Application layer filtering:** NSGs may also be used to restrict traffic based on application-layer protocols. If an HTTP-based DDoS assault is identified, for example, NSGs can be used to restrict traffic from specified HTTP user agents.
- **Traffic analytics:** Traffic analytics can be utilized to detect trends and abnormalities in network flows. This can aid in the detection of DDoS assaults by spotting anomalous spikes in traffic or traffic patterns that are common in DDoS attacks.
- **Auto-scaling:** NSGs may be used to activate virtual machine auto-scaling depending on network traffic. When traffic surpasses a specific threshold, NSGs can deploy extra virtual machines to accommodate the increased demand. By dispersing traffic over numerous virtual computers, this can help avoid DDoS assaults.

4.3.4. Why do they fail in cloud computing?

Although firewalls and network security groups (NSGs) are successful at providing basic security controls in cloud computing environments, they can fail to detect and mitigate distributed denial of service (DDoS) assaults for a variety of reasons.

- DDoS assaults create a massive quantity of traffic, quickly saturating network bandwidth and surpassing the capability of firewalls and NSGs. Even if the firewall or NSG detects the assault, it may be unable to handle the volume of traffic and hence fail to successfully mitigate the attack.
- Firewalls and NSGs are frequently installed in cloud settings as virtual appliances or instances. During a DDoS assault, these virtual instances' resources and processing power are swiftly depleted. When virtual appliances are overwhelmed, they may cease to function, potentially resulting in a security risk.
- Restricted visibility: Firewalls and NSGs normally check traffic at the network's perimeter, where it enters and departs. DDoS assaults, on the other hand, can come from a variety of sources and take diverse shapes, making it impossible to identify and neutralize them with perimeter-based security measures.
- IDS and IPS can sometimes misidentify legitimate activities as malicious, resulting in a False Positive (FP) that can negatively impact web users if disregarded or rejected. In complex network topologies, false positives pose a risk as they may cause IDS/IPS to block non-malicious traffic leading to negative consequences for web users.

- DDoS assaults are continually developing, and attackers are always looking for novel ways to avoid detection and mitigation. Firewalls and NSGs may be ill-equipped to deal with the most recent DDoS attack tactics, leaving them exposed to such attacks.

4.4. Detecting DDoS Attacks Using Machine Learning

The use of supervised machine learning to identify DDoS attacks can be categorized into two types based on their deployment location: source side and destination side. Destination side detection involves deploying DDoS detection models on the cloud servers of the victim. This method is less expensive and easier to implement, providing accurate detection results by aggregating and classifying all received traffic. However, the main disadvantage is that attacks cannot be detected until they reach the victim's system, potentially compromising it.

In recent years, cloud systems have become sources for DDoS attacks on other systems. Attackers can take advantage of various virtual machines to launch attacks on any system. The main advantage of these systems over destination DDoS detection is that they can filter out malicious traffic before it even reaches the victim. However, detecting attacks becomes equally difficult and expensive since cloud systems do not have access to all traffic. In the subsequent section, we will elaborate on the utilization of machine learning algorithms for identifying DDoS attacks, the process employed, and the application of these techniques.

4.4.1. Detection from Cloud Source

More often than not, the source of many attacks originates from virtual machines on a cloud server that is being remotely controlled or rented by attackers to maintain anonymity. As such the detection of the exact source may be difficult to ascertain and protect against due to the difficulty in tracing these machines. In addition to harming the victim, the cloud servers being used to launch attacks can also harm the reputation of the company that owns the servers that are being rented. Because of the damage possible to the companies, source side detection studies have been conducted to traceback these attacks and cut their connections in hopes of not only passively defending against them, but to actively prevent them from harming victims.

Some source side detection providers include; D-Ward[2], a system that monitors inbound and outbound traffic[3]; Multi-Level Tree for Online Packet Statistics, to check the rate of traffic[4]; and MANAnet’s reverse firewall, which prevents attack packages from going out of a cloud[5]. By scanning for these abnormal behaviors in the traffic between the cloud and outside world, we can prevent DDoS attacks of

multiple varieties before they cause too much irreparable damage to their victims and to the companies themselves.

In a study on source side detection[1] several attacks and how to prevent them were detailed. The attacks within the article included SSH brute-force, ICMP flooding, DNS reflection, and TCP SYN attacks. These attacks were chosen due how common they are, specifically in regards to when dealing with DDoS attacks in particular. Although these are the only attacks simulated within the article, similar methods can be extended to look for other types of DDoS attacks by changing the statistics monitored during the events of such attacks.

These attacks were done on virtual machines all with a cloud. Additionally, these attacks were done to simulate both DDoS and DoS attacks by either. The algorithms were also used to differentiate which of these are responsible, by determining if the attacks originated from multiple sources or for a single source.

4.3.2.1 Detecting different DDoS Attacks from the Source Side

SSH Brute-force

Here we will give examples of the attacks and how they can be detected by the server before they harm victims. SSH Brute-force functions and how to detect it are as follows: SSH Brute-force attacks are done by taking control of a victim's computer remotely and using brute-force to guess the passwords of the user. To combat this specific attack, the approach is usually to limit the amount of attempts possible by a user within a timeframe and disconnect them should they continue. However, this can be misused and easily bypassed by simply connecting to the machine again after being disconnected.

In order to stop attacks a machine learning algorithm can be used to monitor the Diffie-Hellman key exchange packages that are generated each time an SSH connection is made. Should multiple keys be generated, the algorithms should be able to determine the connections are originating from the same location repeatedly. Additionally, machine learning can be used to determine if the frequency and the rate of Diffie-Hellman keys are abnormally high. Figure 1[1]. If this information is collected we can assume an SSH Brute-force attack is taking place and prevent further connections to the machine before the password is cracked through the brute-force method applied.

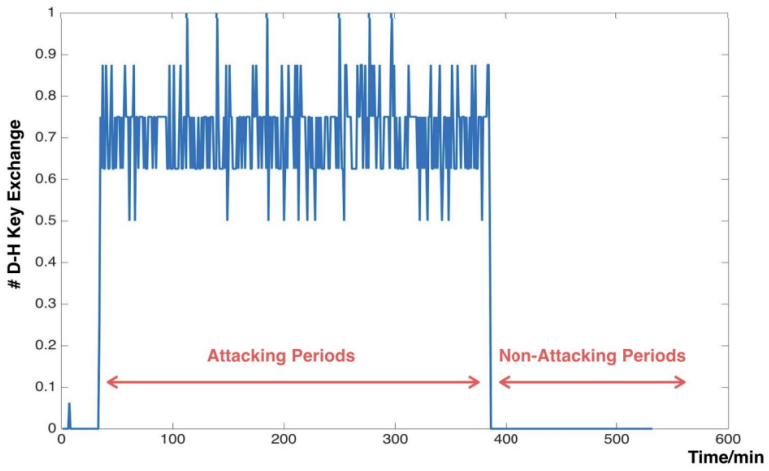

Fig. 3: *SSH Diffie-Hellman key Exchange features[1].*

DNS Reflection

Another example of an attack that can be detected on the source side is DNS Reflection, however another method is required to achieve this. A DNS Reflection attack is done when an attacker sends a large amount of DNS requests to the DNS servers with the victim's source IP spoofed to the request packages. These requests will then ask for heavy-load responses back into the victims own machine. Between multiple requests and their heavy responses, the victim's machine will be overwhelmed and their resources expended.

This attack can be detected by monitoring the inbound and outbound traffic with machine learning algorithms. These algorithms will specifically search for an abnormal ratio of request and response packages. An attack of this kind will result in more request packages than response packages due to the source IP address being spoofed to the victims own. Under normal circumstances, the ratio will be very close to one another. Thus if a large discrepancy is detected, a DNS Reflection attack can be assumed. Figure 2. Once the attack is noticed by this method, the connection from that machine out of the cloud can be severed to prevent the victim from being harmed.

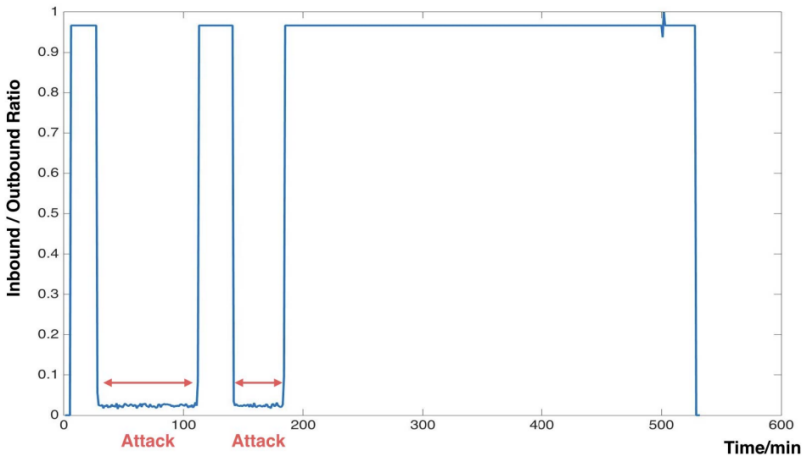

Fig. 4: *Inbound/outbound DNS packages ratio[1]*

ICMP Flood Attack

ICMP Flood attacks are the result of an attack sending a large amount of ICMP packages to overwhelm the victim's machine. Fortunately for its detection we can simply use the rate of these packages in and out of the cloud space to detect an attack. Under normal circumstances ICMP packages are not sent out as often as SSH or DNS, so by using machine learning to find a large spike of ICMP packages being sent within a small time frame, we can conclude that an ICMP Flood attack is taking place and prevent the DDoS attack from harming the victim.

TCP SYN Attack

The last example is a TCP SYN attack. In this type of attack, the attacker abuses the method in which TCP protocols interact via handshakes from one machine to another. Three handshakes are needed when establishing a new connection. First is the original machine SYN x to the server requesting a connection, followed by the server relaying with an ACK x and SYN y to indicate that the request was received and is ready to synchronize. Lastly the client sends an ACK y package to the server as an acknowledgement and to say that it is also ready for synchronized communication.

In the case of a TCP SYN attack, the attacker will send out the initial SYN to begin the process, but when it comes to the third step, the attacker will not reply with the expected ACK y. During this process, resources are being allocated which could be used on other SYN acknowledgements. If multiple requests without ACK responses are made the victim will be unable to connect to legitimate requests since their resources are being consumed by the attacker.

Another version of this attack involves spoofing the victims own IP address so that the victim requests ACK y's from itself, however without the initial request the ACK can not be responded to and the request will eventually timeout, using resources in the process.

We can identify the ratio of TCP packages with the use of machine learning algorithms to help detect an attack. Should the number of SYNs be substantially larger than ACKs then we can determine that an SYN attack is likely taking place. This is because SYNs are typically only used when connections are being established, unlike ACKs which are used in other operations. As such, under normal circumstances, SYN tags should be considerably lower than ACKs. Figure 3.

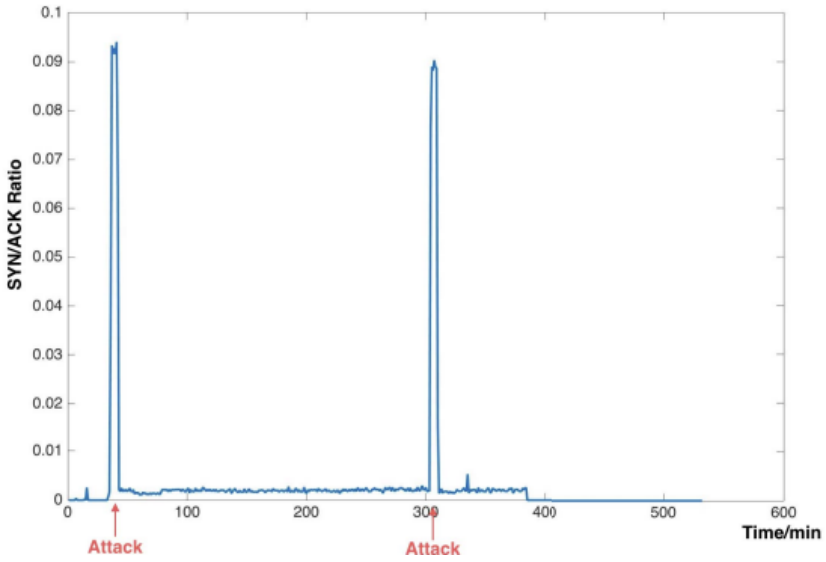

Fig. 5: *SYN/ACK ratio [1].*

4.3.2.1 Algorithms used to detect from Source side

For the machine learning algorithms used in the experiment, there were two types[1]. The first type were supervised algorithms and they included; linear regression, SVM, Decision Tree, Naive Bayes, and Random Forest. The second type of algorithms used were unsupervised. These were K-means and Gaussian-mixture models for Expectation-Maximization algorithms.

The four different attacks were launched from virtual machines within a cloud server to a victim's virtual machine on another server. The detection and defensive system were located on the server where the attack was originating from. These attacks simulated both DDoS and DoS attacks on the victim.

JOINT DETECTION RESULTS OF THREE VIRTUAL MACHINES

Method	Accuracy(%)	FP(%)	FN(%)	Precision(%)	Recall(%)	F1- Score
LR	97.77	0.37	3.82	99.68	96.18	0.9790
SVM Linear Kernel	99.73	0.068	0.44	99.94	99.56	0.9975
SVM RBF Kernel	98.15	3.78	0.24	96.93	99.76	0.9832
SVM Poly Kernel	99.13	0.40	1.27	99.66	98.73	0.9920
Decision Tree	99.07	0.061	0.0167	99.95	98.33	0.9913
Naive Bayes	98.47	3.07	0.27	97.51	99.73	0.9861
Random Forest	99.53	0.00	0.09	100.0	99.12	0.9956
K-means (Unsupervised)	87.76	0.44	22.05	99.54	77.95	0.8743
Gaussian EM	66.53	13.17	50.37	81.94	49.63	0.6182

Table 1: *Accuracy scores from experiment with DDoS attacks.[1].*

The results of the experiment and the respective algorithms utilized are shown in Table 1 [1]. As you can see, the study gathered the accuracy, the False positives(FP), the False Negatives(FN), the Precision, the recall, and the F1-score of each test. The supervised algorithms achieve greater than 93% accuracy and over 0.95 F1 scores. The unsupervised algorithms however did not perform as well, with only 87.76% accuracy and f1 score of 0.87 for k-means.

Different attacks have different identifiers to detect them, especially within a cloud server. Monitoring traffic between the machines and the outside world should help stop potential attacks by those who rent out virtual machines. Additionally, understanding how to utilize machine learning algorithms regarding different types of attacks from the source side should prevent these attacks from reaching out and affecting victims. Security against DDoS attacks regarding cloud servers is a two-front battle, the destination and the source of the attacks. Successful security of a cloud system stopping DDoS attacks will both protect the server owner's reputation and the security of potential victims.

4.4.2. Cloud as a Destination

Since adoption of cloud computing is becoming increasingly widespread, many DDoS attacks see cloud infrastructure as the victim. Because of the wide adoption, it becomes vital to ensure that the cloud infrastructure is protected against such attacks and the detection of these attacks is carried out as accurately as possible. There are various papers that approach this problem by classifying attacks based on the layer of the communication network, mainly transport layer and network layer.

In the review conducted by A. M. Makkawi et al. [39] they evaluated the most popular machine learning algorithms which can be used to detect DDoS Attacks and the current direction in which the research is

progressing. Through their research they found out Support Vector Machines(SVM), Decision Trees, Naive Bayes and Artificial Neural Networks are the most predominant and commonly used algorithms.

The Naive Bayes classifier is an effective and simple algorithm used in DDoS detection. It uses Bayes' theorem to calculate the probability of an event based on prior knowledge of related conditions. Although the algorithm assumes that data attributes are independent of each other, it is still useful in practice. In the DDoS detection context, P(X) represents the initial probability of hypothesis H, while P(X|H) represents the probability of observing packet X given that hypothesis H holds. By using Bayes' theorem, we can calculate the probability of hypothesis H given packet X, which is denoted as P(H|X).

$$P(H|X) = P(X|H) * P(H) / P(X).$$

Decision Trees is a popular machine learning algorithm that constructs a tree-like model to make decisions based on the input features. The algorithm divides the data into smaller subsets based on the feature that offers the most information gain and continues this process recursively until the leaf nodes are pure. The process of building the decision tree involves the following steps:

1. Choose the attribute with the highest gain ratio and use it to split the data into different branches, each representing a possible value of the selected attribute.
2. Group the instances in the training set based on the chosen attribute.
3. Repeat steps 1 and 2 for each branch. Our implementation involves defining four classes, including normal, TCP SYN flooding, UDP flooding, ICMP flooding, and attributes derived from traffic signatures. The training set is used to construct the decision tree, which is then used to classify incoming traffic.

Support Vector Machines (SVMs) are a type of supervised machine learning algorithm that finds the hyperplane that best separates the data into different classes. They use support vectors to define the hyperplane, which can handle datasets with high dimensionality and are effective in dealing with non-linearly separable data. SVMs are less prone to overfitting than other models but can be computationally expensive. For DDoS Detection of the following TCP SYN, HTTP and UDP packet floods it can be reduced to a multiclass problem with support vectors dividing the feature space into sub regions.

Machine learning algorithms like the random forest can be applied to a variety of tasks like classification and regression. For various subsets of the data, numerous decision trees must be created, and their predictions must then be combined to get a final prediction. By randomly picking subsets of the data to

train each decision tree, this method combats the problem of overfitting. Depending on the issue at hand, either voting or averaging can be used to determine the outcome.

Artificial Neural Networks (ANNs) are a machine learning algorithm inspired by the human brain. ANNs consist of interconnected nodes, called neurons, which process information through weighted connections. ANNs can handle complex and non-linear relationships in data, but are prone to overfitting and can require a large amount of data and computational resources for training.

Zargar et al. [25] discusses machine learning techniques based on identifying features present in packet headers. One such defense is the HOP count filtering which is used to limit the propagation of packets in a network by limiting the number of "hops" or intermediate devices a packet can pass through before being discarded. This helps prevent packet loops and reduces unnecessary network traffic. Essentially, hop count filtering sets a maximum limit on the number of devices a packet can traverse before it is considered to have reached its destination or to be lost. This helps destination filter out packets from altered source IP addresses from unaltered ones. IP spoofing attack is another such attack discussed in[34]. The attacker either enters a legitimate IP address or an unreachable IP address. This way, the attacker intercepts the communication or the server is kept busy by trying to reach an unreachable IP address. Hop count filtering is a way to overcome this attack.

The paper goes on to describe further assaults and often employed defenses against them. In a smurf attack, for instance, the attacker sends several ICMP echo requests with broadcast IP addresses as the target address. In order to guarantee security, operating systems at PaaS levels must deny ICMP messages with the broadcast IP address as the destination. SYN flooding is a dangerous transport-layer attack that takes place when an attacker sends SYN packets incessantly without ever receiving an ACK packet back in response. In a sniffing attack, the attacker sends a packet with the anticipated sequence number of an active TCP connection using a fictitious IP address. Because of this, the server cannot respond to that request, which has an effect on the cloud system's resource performance. Significant issues with the planned defenses include decreasing performance of the cloud system and increasing latency.

Another approach of classification is presented in [35]. They suggest signature-based detection, anomaly based detection and stateful protocol analysis. Signature based detection includes creating a database of known attacks by tracking events and patterns. When a new attack matches these signatures, the traffic is detected as an attack and is stopped. Anomaly based detection, on the other hand, seeks to define normal traffic and any traffic which doesn’t match this definition is deemed as an attack and handled accordingly.

SPA is a more general approach to defining this profile considered to be normal traffic but it works with different protocols to come up with this definition.

The pre-processing module formats gathered packets by removing extraneous data that has a weak connection to detection, as shown in the figure. It effectively recognizes known threats by comparing a specific network event with the rules stored in a knowledge base, much like signature-based detection does. The advantage of this approach is that it enables us to update the knowledge base without altering the existing regulations. A learning base is used to collect and store information about the behaviors of authorized users over time. The validity of the user is then determined by using a machine learning algorithm to this data. C4.5 is a decision tree based machine learning algorithm that has been widely used in this area.

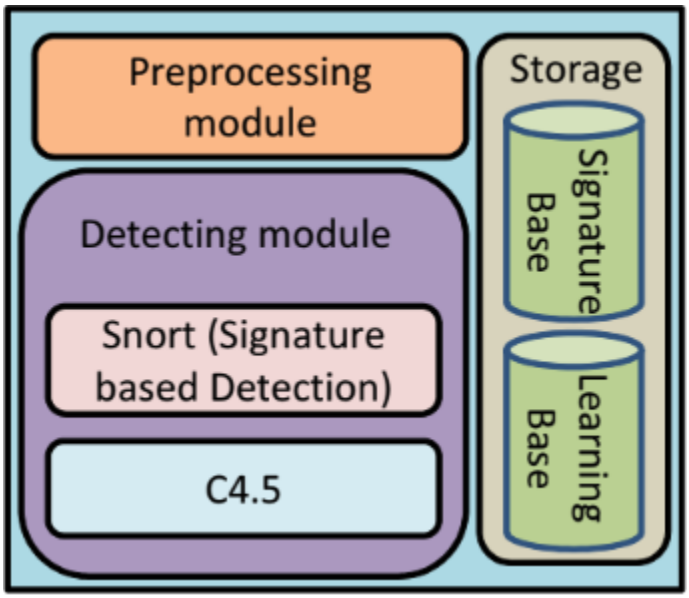

Fig. 6: *Proposed methodology to use multiple strategies to detect attacks*[4].

A different research paper[36] compared the effectiveness of C4.5 with other machine learning algorithms. The study involved using the SVM classifier, a K nearest neighbor algorithm, and C4.5, and discovered that C4.5 outperformed the other methods in terms of precision and sensitivity. The experiments were conducted using the NSL-KDD dataset, which was created by an intrusion detection system based on network traffic. The dataset contains 41 features and 2 labels that indicate the type of traffic.

A comparative study between various algorithms was conducted in experiments performed in [38]. The authors create a cloud environment and initiate attacks using the Tor Hammer tool. Using the attacks they

designed, the authors created a dataset consisting of 10 features (time, duration, source ip address, protocol, source port, destination ip address, destination port, number of packets, number of bytes) was used to train Support Vector Machines, Naive Bayes Classifier and Random Forest, among which SVM performed best in terms of Recall, Precision, Specificity, Accuracy and F1 measure.

	SVM	Random Forest	Naïve Bayes
Recall	0.998	0.993	0.860
Precision	0.998	0.992	0.881
Accuracy	0.997	0.976	0.980
Specificity	0.996	0.995	0.505
F measure	0.998	0.996	0.826

Table 2: *Comparison of accuracies achieved for various machine learning algorithms.*

A slightly different process flow is proposed in which the authors [37] preprocess the data collected using Apache Spark, which is a distributed computing solution. Distributed computing allows for faster preprocessing. The processed data is then used to train a deep learning model to achieve 99.73% accuracy. A two flow process where a parallel ensemble model is used along with a CNN based lightweight model is used to divide flows into spatial and temporal features respectively. This system provides 98% accuracy and 97% precision.

4.5. Problems with Machine Learning for DDoS Detection

For supervised machine learning algorithms they need enough labeled sampled data which is representative of all the different kinds of DDoS attacks so that the models can be trained and implemented. The major problem faced for research in this field is lack of such data. Most of the research and models have considered datasets NSL-KDD, CICIDS2017 and CIC-DDoS2019 for training and testing the models

Another problem for supervised machine learning algorithms is that the amount of data generated within the cloud is very large. Supervised machine learning may not be able to handle such large amounts of data and may not provide proper classification with a bad accuracy score. Also in conjunction to the large data size the number of dimensions or features to the data are also very high. CIC-DDoS2019 contains 80 features from which the best features that provide the most distinctive information have to be selected for all the algorithms. CICIDS2017 contains 86 features. Thus we have to optimize these algorithms so that they can work efficiently in cloud systems.

4.6. Optimizing Machine Learning Algorithms

4.6.1. Forward Feature Selection

Forward Feature Selection (FFS) is a technique used in machine learning to select the most relevant features from a dataset for a given task. The FFS algorithm works by iteratively selecting a subset of features that improve the performance of the model. In each iteration, the algorithm adds a new feature to the subset and evaluates the performance of the model using cross-validation or other validation techniques. If the performance improves, the feature is kept in the subset, and the algorithm moves on to the next iteration. If the performance does not improve, the feature is removed, and the algorithm terminates. FFS is particularly useful in high-dimensional datasets, where the number of features can be larger than the number of samples. By selecting a subset of the most relevant features, FFS reduces the dimensionality of the input data and improves the efficiency and accuracy of the model.

The paper by Lonnie et.al.[26] proposes a system for detecting (DDoS) attacks on the cross-plane of a Software Defined Network (SDN) using a hybrid deep learning approach. The proposed system consists of a Forward Feature Selection (FFS) algorithm and a hybrid deep learning model that combines Convolutional Neural Networks (CNNs) and DNN(DNNs) networks. The FFS algorithm is used to select the most relevant features from the network traffic data, reducing the dimensionality of the input data and improving the efficiency of the deep learning model.

The study employed the CICIDS2017 dataset, which comprises 86 network features, including various types of attacks such as botnet, port scan, DDoS, and web attacks. Deep learning is an AI function that enables fast detection and classification of network attacks. The researchers proposed a hybrid approach using Convolutional Neural Networks (CNNs) and Deep Neural Networks (DNNs) to develop an IDS for control and data planes. The first step involves preprocessing the dataset to remove errors such as unwanted elements, infinite values, redundant features, and missing values, which resulted in 20 out of the 86 features being retained. A CNN and DNN is created which in turn is trained with the 5 features selected from Forward Feature Selection.

Through this it was found that these features the best output for DDOS detection: ‘Fwd IAT Max’, ‘Flow IAT Max’, ‘Idle Max’, ‘Avg Packet Size’, ‘Total Backward Packets’. These features when used for the CNN and DNN model offered an accuracy rate of 98.09%, specificity 97.93%, false-positive rate 0.0207% &, recall 98.32%, precision of 97.31% and F1-score 97.81%.

While the model was trained and tested for Software Defined Networks, the same model can work in cloud systems by dividing the cloud architecture into Control Plane and Data Plane as with SDNs and the same algorithms can be implemented in cloud systems.

4.6.2. Supervised Machine Learning in conjunction with ANN

A Feedforward Neural Network (FFNN) is a type of artificial neural network that is organized in layers, with each layer consisting of a set of neurons that process inputs and produce outputs. In an FFNN, information flows only in one direction, from the input layer, through one or more hidden layers, to the output layer. FFNNs are commonly used for supervised learning tasks such as regression and classification. They are efficient models that can capture complex relationships between inputs and outputs.

In supervised learning, the aim is to develop a function of inputs to output mapping based on a set of labeled training samples. Multilayer Perceptron (MLP) is a supervised learning algorithm. This means that during the training process, the MLP is provided with a set of labeled training examples where the input and corresponding output values are known. The MLP then learns to map the inputs to the correct outputs by adjusting its internal parameters or weights. Alkasassbeh et.al [8] establishes that MLPs are particularly effective at this task because they can learn non-linear relationships between inputs and outputs and can handle high-dimensional data. MLP has been utilized in many applications, including time series prediction problems, classification, regression and simple auto-regressive models as mentioned in [9] [10].

As part of the study conducted by Mouhammd Alkasassbeh et.al [11], A Machine Learning classifier is researched and evaluated. As there are no proper data sets that include novel DDoS attacks and as other available data sets may include many duplicate and redundant records that may result in unrealistic outcomes, Evaluation is carried based on a new dataset collected which contains four types of DDoS attacks as follows: (SIDDoS, UDP Flood, HTTP Flood and Smurf) without duplicate and redundant records.

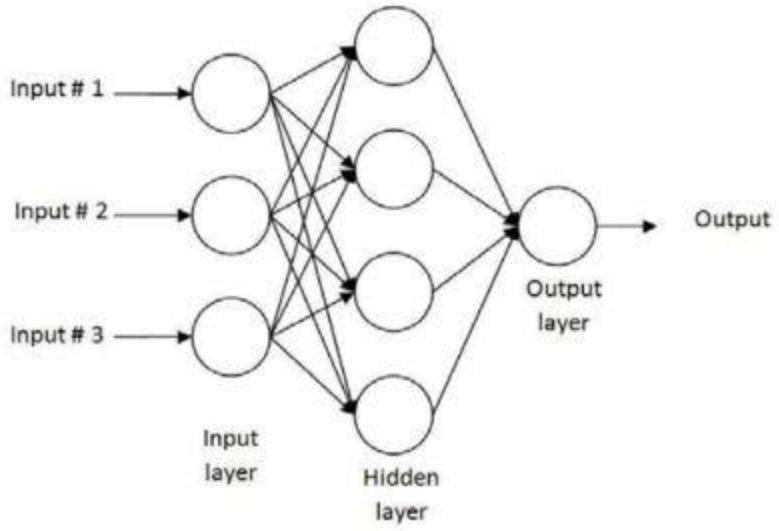

Fig. 7: *Multilayer Perceptron Architecture with a single hidden layer*

MLP networks can be distinguished based on three main performance characteristics as specified by Krse et. al and Fausett et. al in [12][13]:

- The first characteristic is the Neural Network Architecture, which refers to the pattern of connections between neurons in multiple layers. Each layer has two nodes that are connected end-to-end, and MLP is always fully connected. The weights of each link are limited based on the training algorithm, and more complex architectures have additional layers.
- The second characteristic is the Training Algorithm, which involves selecting one model from a set of models and determining the weights of the connections.
- The third characteristic is the Transfer Function, which is applied by each neuron to its net input to determine its output signal. This function is typically non-linear.

The MLP (Multi-Layer Perceptron) can be customized by specifying the following parameters:

- The number of neurons in the hidden layer, which is set to 16
- The learning rate, which is set to 0.3
- The momentum, which is set to 0.2
- The maximum number of epochs, which is set to 500.

This algorithm was trained on the dataset using 66% of the data collected and 34% were used as test data. To evaluate and analyze the performance of the classifiers, A confusion matrix approach is employed based on the evaluation metrics listed in [14]. A comparative analysis was also done against Naive-Bayes

classifier and random Forest algorithm. The below image depicts the resultant confusion matrices for MLP. Using this matrix, the accuracy, precision, and recall of the models were computed. The overall accuracy for MLP, Random Forest, and Naïve Bayes was found to be 98.63%, 98.02%, and 96.91%, respectively.

	Normal	UDP Flood	Smurf	SIDDOS	HTTP -Flood
Normal	657961	0	0	70	20
UDP Flood	6767	61765	0	0	0
Smurf	2817	0	1396	132	10
SIDDOS	115	0	0	2136	0
HTTP -Flood	0	0	0	86	1352

Table 3: *Evaluation of MLP based on Confusion matrix*

MLP exhibited better precision and recall outcomes for minority classes, with Naïve Bayes performing the worst. The Smurf class posed significant challenges for all classifiers, owing to its technique of sending a large volume of ICMP echo packet requests, which are difficult to classify as normal or anomalous traffic. ICMP complements the traffic management of IP4 and missing flow control, resulting in poor detection rates for Random Forest and Naïve Bayes for the Smurf class. On the other hand, MLP reported high precision rates. Overall, the findings suggest that MLP is the most effective classifier for identifying DDoS, with promising performance outcomes.

In MLP, ANN continually learns until the error minimization criteria is met.. Assuming that the required output is P and the resultant ANN outcome is P`, the learning process will only stop only when the error difference between them is minimal. The aim of the training process is to find the set of network weights that cause the result of the network to match the exact values closely. However, we must not forget that the MLP always takes the longest time for training [8][9][10].

4.6.3. Hybridized Machine learning IDS models

In order to examine the network traffic and spot abnormalities or patterns that are consistent with a Ddos attack, Intrusion Detection Systems (IDS) are useful in the detection. IDS can track incoming network data and spot patterns that don't match up with ordinary traffic. For example, Rule-based IDS can look for an abnormally high number of requests from a single source, or a sudden surge in traffic from multiple

sources. IDS can also identify DDoS attacks by using machine learning algorithms. These algorithms can be trained on labeled data to discover the traits of typical traffic and spot outliers.

DDoS attacks can be challenging to detect and mitigate because the traffic appears to be legitimate, but it is coming from many different sources at the same time. Also, As per Muraleedharan et.al [16] DDoS attacks are highly dynamic and constantly evolving, making it difficult for traditional machine learning based IDS models to learn from a fixed set of attack patterns. It is a challenging task to differentiate between legitimate and illegitimate traffic. This is because obtaining labeled data for training models is challenging due to the rarity of DDoS attacks. Attackers are constantly changing their techniques to evade detection, resulting in overfitting and false positives. Additionally, limited visibility into network traffic can make it complicated for machine learning models to capture all aspects of a DDoS attack. Therefore, while machine learning models have the potential to detect DDoS attacks, Sumathi et. al [15] transcended further and used the models in conjunction with other supervised ML classification techniques such as Artificial Neural Networks (ANN), Support Vector Machines (SVM), and Decision Trees (DT), to create a robust and accurate hybrid IDS model that brings the allround benefits from each of the IDS models.

Machine learning algorithms such as C4.5, SVM, and KNN are all utilized for classification of attack data in IDS models, but their suitability and results vary. The decision tree-based C4.5 algorithm divides the data into subsets that are as homogeneous as possible in terms of the target variable using an information gain measure and SVM (Support Vector Machines) is a supervised learning algorithm that separates the data into different classes using a hyperplane. KNN (K-Nearest Neighbors) is a non-parametric algorithm that classifies new data points based on the class labels of the k-nearest neighbors in the training data. KNN uses a distance metric to measure the similarity between data points, and the value of k determines the number of neighbors to consider.

The proposed hybrid model follows a two-stage approach. In the first stage, the network traffic data is preprocessed, and feature selection is performed. The feature selection algorithm helps in selecting the most relevant features from the dataset, thus reducing the dimensionality and improving the model's performance. In the second stage, the selected features are fed to the classifier model, which is composed of ANN, SVM, and DT. Each classifier is trained on the same dataset, and the final output is obtained on all these three classifiers using a weighted averaging method.

The IDS model shown below is constructed with Dataset Preprocessing K-Fold validation Classifier and experimentally verified with a 10-fold unknown test dataset.

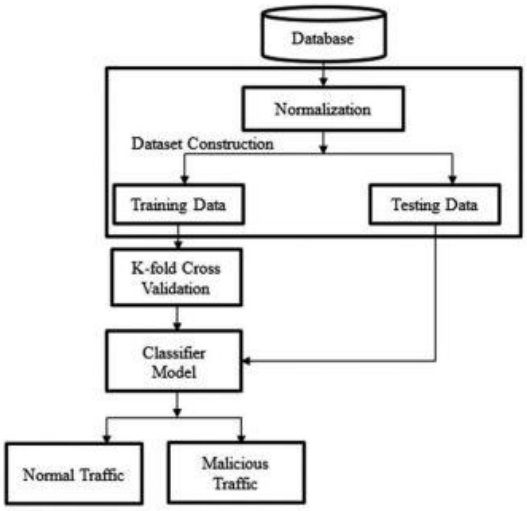

Fig. 8: *Representation of the ML based intrusion model as a block diagram*

As per the evaluation metrics shown in Fig. derived from experimental verification on the NSL-KDD dataset, The classic SVM classifier model reported better accuracy, but the precision and sensitivity of the C4.5 classifier algorithm is better than that of SVM and KNN models.

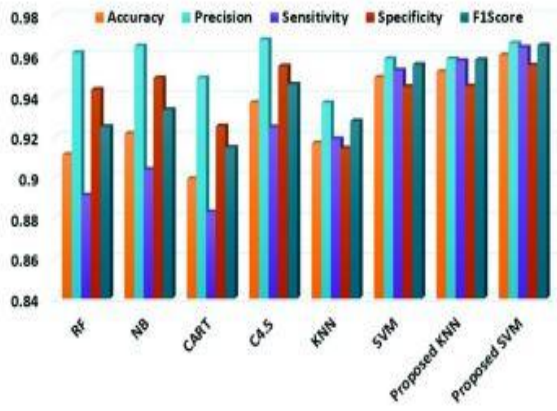

Fig. 9: *Comparison of Proposed ML based IDS with existing systems*

The selected attributes of the C4.5 classifier algorithm are fed into the SVM and KNN classifier algorithm, and the corresponding performances are analyzed. A Hybrid SVM classifier based IDS model with the features selected by C4.5 classifier algorithm has overtaken the traditional methods. The model's performance was evaluated using various performance metrics such as accuracy, precision, recall, and F1-score. The results showed that the proposed hybrid model outperformed the individual classifiers in terms of accuracy and other performance metrics, indicating its effectiveness in detecting DDoS attacks. The integration of these two algorithms has got the advantages of both algorithms, which leads to

providing better results than the conventional algorithms. Therefore, The hybrid model's accuracy and robustness make it a promising solution for detecting and mitigating DDoS attacks, thus contributing to the overall security of computer networks.

4.6.4. Perplexed Bayes Classifier

Despite its effectiveness, Naive Bayes faces some challenges in real-world applications. One of the primary challenges is the assumption of feature independence, which may not hold in complex datasets. This can lead to inaccurate classifications and reduced performance. Another challenge is the handling of missing data, which can affect the accuracy of the classification. Additionally, Naive Bayes can be sensitive to outliers in the data, which can skew the classification results.

To address some of these challenges, the Perplexed Bayes classifier algorithm was developed by Carlos et.al [7] . The algorithm is an extension of Naive Bayes that addresses the issue of feature independence. Perplexed Bayes takes into account the correlations between features by using a perplexity measure that estimates the complexity of the data distribution. This allows for more accurate classification results in complex datasets.

Another key advantage of Perplexed Bayes over Naive Bayes is its ability to handle missing data. Perplexed Bayes uses a "lazy learning" approach that only makes use of the available data when making a classification decision. This approach is particularly useful when dealing with sparse data, which is common in many real-world applications.

The research article by Mishra et.al.[6] proposes a perplexed-based classification algorithm that is more efficient and accurate in detecting DDoS attacks on cloud systems, compared to existing ML algorithms like naive Bayes and Random Forest, and nature-inspired feature selection like genetic algorithm (GA) and particle swarm optimization (PSO).

To explain about the Perplexed Bayes classifier, we first explain about the Naive Bayes classifier.
Naive Bayes classifier is a probabilistic algorithm that makes classifications based on the Bayes theorem. It assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

The formula for Naive Bayes classifier is:

$$P(C|X) = P(X|C) * P(C) / P(X)$$

where C is the class variable and X is the feature vector.

The Perplexed Bayes classifier uses the reciprocal of perplexity (the geometric mean) as the combination operator for individual feature probabilities.
The formula for Perplexed Bayes classifier is:

$$P(C|X) = (1 / (\text{perplexity})) * \prod P(Xi|C)$$

where C is the class variable, X is the feature vector, Xi is an individual feature and perplexity is defined as:

$$\text{perplexity} = \prod (1 / P(Xi))$$

What the Perplexed Bayes classifier does mathematically is that the naive independence assumptions disappear. Constructing this novel classifier is as effortless as creating a Naive Bayes classifier, exhibits identical precision, acquires knowledge rapidly, and provides much-improved confidence scores.

The research by Mishra et.al.[6] involved a dataset containing various DDoS attacks and their corresponding features, from which important features for detecting DDoS attacks on cloud computing were identified based on correlation. These selected features were then incorporated into the proposed algorithm for training.

The Perplexed Bayes classifier was trained on the NSL-KDD data set, which consists of 43 features dealing with traffic input and label and severity of the attack. The data was first preprocessed and null values were removed. Then, features were selected based on their correlation with the target value, from which 20 features were extracted. The labels were then converted from multiclass to binary labels. The data set was then split into 70% for the training set and 30% for the test set and then trained with the extracted features, with a label of 1 for denoting DDoS, otherwise 0.

The perplexed classifier with feature selection, performed better than the other classification algorithms (naive Bayes and Random forest), and was better than nature-inspired feature selection (Genetic Algorithm and Particle Swarm Optimisation) algorithms in terms of Accuracy, Sensitivity and Specificity. The perplexed classifier without feature selection was better than naive Bayes classifier, but worse than Random forest.

Finally, the paper concludes with the finding that the "service" features need to be more focused on detecting DDoS attacks, since it is the most correlated feature to the target/goal variable.

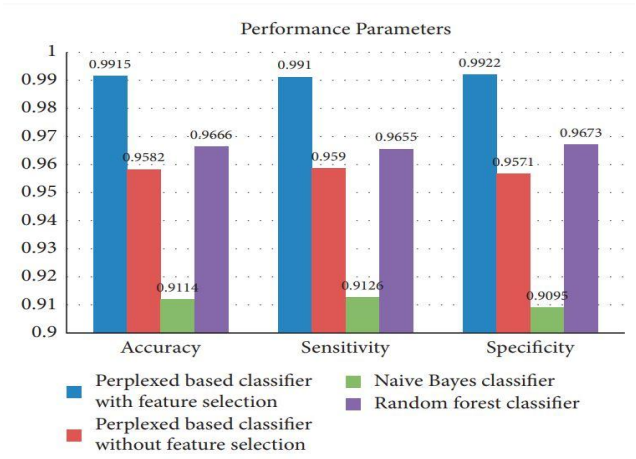

Fig. 10: *Performance parameters of the algorithms (PBC/F, PBC/ WF, NBC, and RF).*

The approach proposed by Mishra et.al.[6] can also be used for any attack on the cloud, where the data characteristics are not interdependent.

4.7. Deep Learning for DDoS Detection

Deep learning is a form of machine learning that is centered on artificial neural networks. It takes inspiration from the structure and function of the human brain to learn patterns and features from vast amounts of data. Its ability to recognize complex patterns and relationships in the data allows it to learn and generalize from extensive datasets.

The core of deep learning is made up of artificial neural networks, consisting of interconnected nodes that take inputs from other nodes and use mathematical functions to produce outputs. The outputs from one layer of nodes become the inputs for the next layer, repeating the process until a final output is generated. Several types of neural networks, such as convolutional neural networks (CNNs), recurrent neural networks (RNNs), and deep belief networks (DBNs), are available for use in deep learning, each tailored to specific data types and tasks.

Deep learning provides a good advantage in case of machine learning models used for security challenges and DDoS detection due to their various advantages such as its ability to extract relevant features. In case

of DDoS detection where the dataset for training models is limited, deep learning can extract the most informative features from the already limited dataset to improve the accuracy of the model.

4.7.1. Detecting DDoS Attacks at Gateway Router using Deep Learning

A gateway router in cloud systems is a device or software that connects a local network to the internet or another network. It acts as an entry point and exit point for traffic entering or leaving the local network. In cloud systems, the gateway router is typically a virtual appliance that runs in the cloud environment and serves as the central point of control for network traffic.

The gateway router in cloud systems provides several functions, including routing, firewalling, and network address translation (NAT). It routes traffic between different networks, such as between a virtual private cloud (VPC) and the internet. It also acts as a firewall, allowing or blocking traffic based on predefined rules, and performs NAT to translate IP addresses between the local network and the internet. It is this particular feature of a gateway router that can be taken advantage of and a machine learning model can be created that would detect whether the traffic is valid or malicious.

All the incoming traffic to the gateway router can be collected and the relevant features extracted from the traffic for a time duration ∇t as proposed in [37]. On these extracted features the trained deep learning model can be run and can be identified as malicious. For each and every packet deemed as malicious the gateway router can drop these packets after a certain time period thus blocking any attempt to deny service to actual users.

The proposed model for this was trained on the KDDCUP99 dataset which achieved 99.5% accuracy for DDoS Detection. Resilient Distributed Datasets were created which can be used for training and testing the proposed architecture as represented in the Fig.

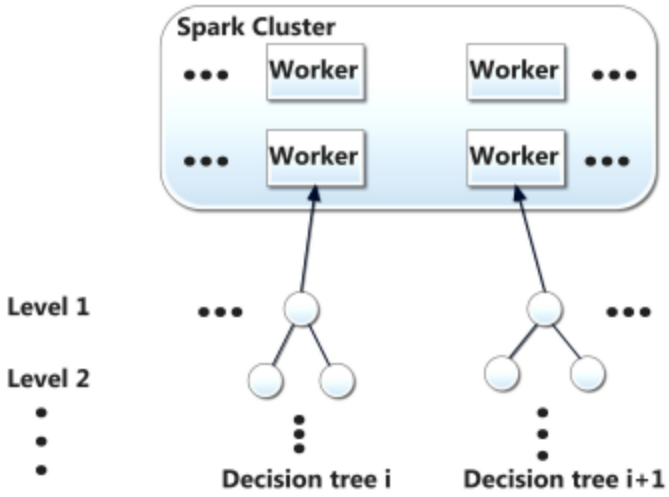

Fig. 11: *Proposed RDD's using Apache Spark*

4.7.2. Detecting DDoS Attacks with a Hybrid Ensemble Model

All the DDoS detection techniques and approaches have been working on a certain set of features that when clubbed together provide enough differentiating power from benign and malicious activity. These features can broadly be classified into two separate categories - spatial and temporal. Spatial features can include network traffic features related to the spatial characteristics of the traffic, such as the source and destination IP addresses, source and destination ports, and protocol. Meanwhile, temporal features can include network traffic features related to the temporal characteristics of the traffic, such as the time of day, duration of the traffic, and the inter-arrival time between packets.

Training two different models based upon spatial and temporal features and then combining the prediction of these two models can provide more accurate classification for DDoS attacks if they are benign or malignant. This basic idea was implemented by the authors Kumbala Pradeep Reddy et al. [17] in their paper with the basic flowchart shown below.

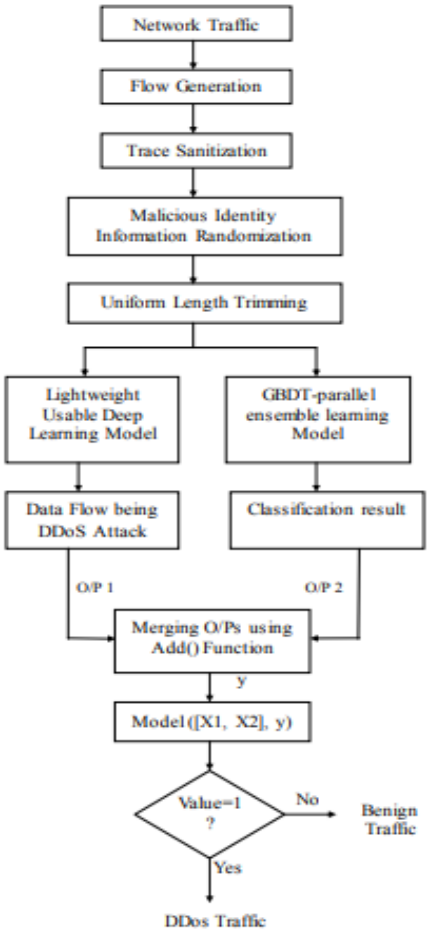

Fig. 12. *Flowchart for the proposed hybrid Architecture*

In the proposed system, the incoming data is first preprocessed and from packets belonging to the same flow, the first few ones are detected and formed as a segment.

Gradient Boosting Decision Tree (GBDT) is utilized to identify DDoS Attacks by employing spatial features. The spatial prediction trains are obtained for each fold in parallel. These spatial features are employed to create spatial prediction trains. Later, all the five parallel spatial prediction trains are merged to form a single spatial prediction train. This trained model is used for making the predictions. At the end a spatial test prediction train is obtained for each fold. The average spatial dataset is calculated to get a train dataset of spatial features.

A lightweight Convolutional Neural Network(CNN) is deployed to classify the network traffic using temporal features. This proposed CNN consists of four layers such as input layer, CNN layer, max pooling layer and the classifier layer. The model extracts the features of network flow automatically by

classifying them into various kinds. The first layer takes the traffic flow as an input which has individual packet vectors. The CNN layer processes each traffic flow through a single convolutional layer, using kernel or sliding window filters to convolve the traffic flow. This is done to extract temporal features that contain important information, which are then learned to classify the network flow as normal/benign or a DDoS flow.

The results from both the models are then combined to form the output of the hybrid deep learning based ensemble framework model. The GBDT model provides a 93% accuracy while the Lightweight CNN acquired nearly a 95% accuracy. When these two are combined to provide the final result of the whole model the accuracy comes up to 98%.

4.7.3. DDoS Attack Detection Method Based on Deep Transfer Learning

One of the major problems that we have discussed is the lack of labeled data for different types of DDoS Attacks. Supervised Machine Learning algorithms cannot be trained to perform efficiently if they do not have enough data. One solution to this problem is to perform deep transfer learning where the model is trained on a specific type of DDoS Attack and then the trained model is transferred to detect another DDoS attack. J. He et al. [18] have proposed this method of deep transfer learning using neural networks.

SYN Flood is the most prominent type of DDoS attacks with 71% of DDoS attacks being either SYN Floods or DNS attacks according to reports published by cloudflare. Another type of DDoS attack which is harmful and emerging is LDAP. LDAP DDoS attack is a type of DDoS attack that exploits the Lightweight Directory Access Protocol (LDAP) to overwhelm a target system with a flood of connection requests.

Authors of this paper first chose a neural network consisting of six fully connected (Dense) layers, with batch normalization and ReLU activation applied after each layer. The input layer has 100 features, and the output layer has two units for classification. The first three Dense layers progressively increase in the number of units, with 500, 1000, and 2000 units respectively. The next two Dense layers decrease in the number of units, with 3000 and 2000 units respectively. The batch normalization layers normalize the outputs of the Dense layers across the batch dimension to help stabilize training and improve generalization, while the ReLU activation function ensures that the output is non-negative. In addition to this model, they also considered 3 other models that are derived from the first model by adding more fully connected hidden layers that have gradually reduced layer width (i.e, number of neurons in a layer).

Coming to transfer learning, it is performed by freezing the weights of the initial hidden layers and proceeding to train the network on another domain. This can save time and resources compared to training a new model from scratch and can be especially useful when there is limited data available for the new task. These neural networks models are trained on SYN attacks and then transferred the model weights for Detecting LDAP attacks. SYN Attacks are of source domain and LDAP attacks are of target domain. The authors chose to not freeze the weights only in the last hidden layer and the transferability performance was calculated. It was found that the 8LANN neural network provided the best classification in the target domain.

Network Name	**Target Domain detection Performance**	**Transferability Value**
6LANN	98.67%	19.47
7LANN	99.18%	19.63
8LANN	99.28%	19.65
9LANN	99.08%	19.60

Table 4: *Comparison of Different neural networks*

4.7.4. Detecting DDoS Attacks using FedAvg and FLAD

In the research paper published by McMahan et al[19]**,** the authors propose a new concept of Federated Learning(FL), a distributed training approach with focus on the privacy of the individual participants in the FL process. Federated Learning involves a central server and a set of K clients, each with a fixed local dataset[20]**.** While this was initially introduced by the authors for Image Classification it can easily be implemented for other fields such as DDoS Detection. For each round of federated learning a certain fraction of clients are selected on whose data model is trained and all the trained models are aggregated by the central server. This aggregated ANN model is then used to calculate accuracy across all the clients from the representative dataset present at the central server. In the next round, clients having a worse accuracy are selected for training and optimizing the current global model. These rounds are repeated till a certain accuracy has been achieved. The proposed procedure could be difficult for several reasons, including having data that is not independent and identically distributed among clients, having imbalanced datasets, and dealing with unreliable communication links between clients and the server.

One of the problems of this proposed model while implementing DDoS detection in the cloud is that the central server should have access to a representative sample of data from all the clients to test the accuracy of the systems. In this particular case, the clients will probably not share the network traffic data which can impact the security considerations and also provide information about the business of the client to competitors. Thus as an optimization to FedAvg algorithm, the Federated Learning Approach to DDoS attack detection (FLAD) was introduced that provides major advantages over FedAvg for DDoS detection such as low convergence time and no exchange of training and validation data[20].

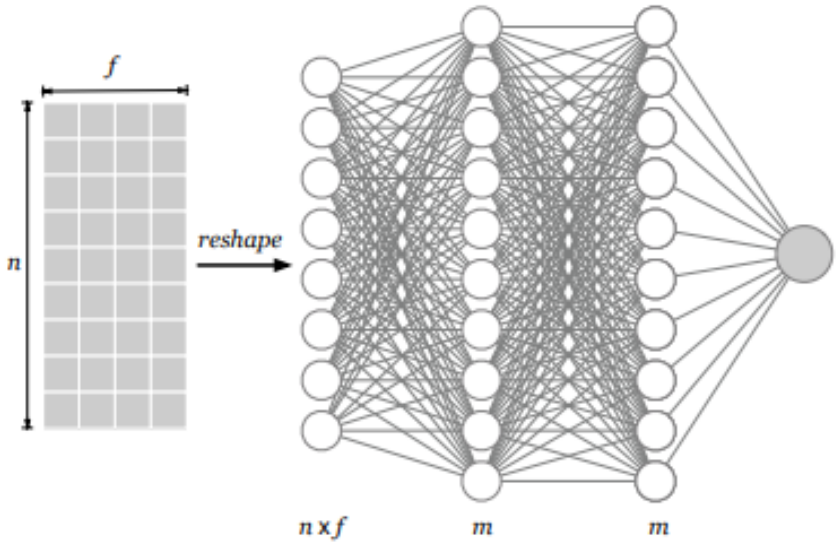

Fig. 13: *Architecture of ANN used to evaluate FLAD*

To train the model, a large number of clients are needed. In their experiment, the authors divided the CIC-DDoS2019 dataset into 13 parts, with each unique DDoS attack targeting a particular client. This resulted in a non i.i.d distribution of the dataset. From the 86 features in the dataset, the authors used Jensen-Shannon Distance (JSD) to identify the most informative features, which were Time, Packet Length, Highest Protocol, IP Flags, Protocols, TCP Length, TCP Ack, TCP Flags, TCP Window Size, UDP Length, and ICMP Type information. The model was trained using these features, specifically the #Flows feature, which indicates the number of bi-directional TCP sessions or UDP streams in the traffic traces provided with the dataset. This feature was preprocessed as arrays with a shape of n=10 rows and f=11, and then fed to the ANN architecture shown in the figure above.

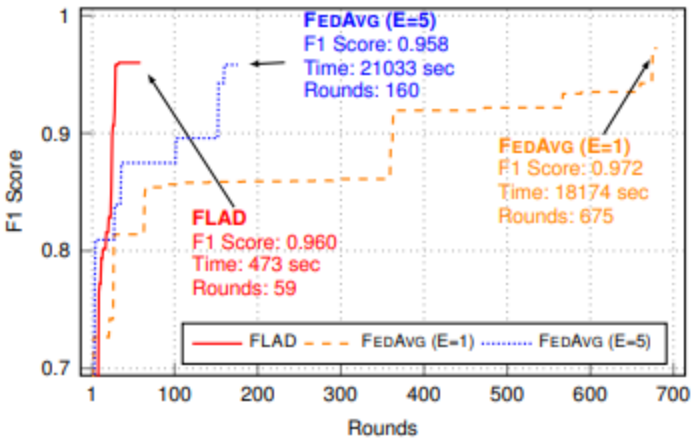

Fig. 14: *Accuracy score for FLAD and FedAvg*

The model is then run in rounds same as FedAvg where certain clients are selected for each round and the ANN model is trained on their dataset. The updated global model is then used to calculate accuracy across all the machines which identifies clients which have to be selected for the next round. If the accuracy does not improve but much in 25 rounds(patience) then the FLAD model is stopped and the final ANN model is created which is implemented by all the clients in their cloud system. As shown in figure above FLAD converges faster with higher F1 Score than FedAvg.

While the F1 score of FLAD does not seem a lot in comparison to previous algorithms discussed, where it truly shines is its ability to detect various types of DDoS attacks with very high true positivity rates. FLAD can detect various different kinds of DDoS attacks with very high accuracy.

Attack	FLAD	FedAvg	
		E=1	E=5
WebDDoS	0.9136	**0.0727**	**0.6364**
LDAP	0.9889	0.9222	0.9306
Portmap	1.0000	0.9058	0.9183
DNS	0.9872	0.7799	0.7852
UDPLag	0.9978	0.9969	0.9978
NTP	0.9770	0.9567	0.9468
SNMP	0.9992	0.9477	0.9578
SSDP	0.9982	0.9973	0.9958
Syn	0.9865	**0.2762**	**0.3243**
TFTP	0.9949	0.9513	0.9242
UDP	0.9987	0.9986	0.9968
NetBIOS	0.9983	0.9085	0.9282
MSSQL	0.9983	0.9863	1.0000
Global	0.9977	0.9443	0.9557

Table 5: *TPR for different types of DDoS attacks with FLAD and FedAvg*

4.7.5. Detecting DDoS Attacks with Hierarchical Temporal Memory

Attackers are constantly developing new types of DDoS attacks and updating known ones, making it important for models to stay up-to-date with the latest information. One approach is to create new models based on new training data, but another option is to train the model based on the data it's classifying and reinforce it. Based on this particular idea Manh-Hung Nguyen et al. [41] proposed a machine learning model based on Hierarchical Temporal Memory which can update itself. Hierarchy Temporal Memory (HTM) is a machine learning algorithm developed by Numenta, which is inspired by the structure and function of the neocortex in the brain. HTM is designed to analyze and learn from time-series data, and it is particularly useful for applications that involve anomaly detection and prediction.

A basic HTM model has three sections - semantic encoder, spatial pooler and temporal memory. The Semantic Encoder converts various input data's feature vectors into binary vectors. Then, the Spatial Pooler transforms the binary vectors from the input space into sparse arrays.The way the Spatial Pooler works enables HTM to keep the input space sparse and maintain overlap. When inputs are similar, they have a high overlap, and when they are different, they have a low overlap.

The incoming data is preprocessed where features that can be calculated in real time are extracted. The inter-arrival time (IAT) of each incoming packet is calculated, and packets with an IAT below five microseconds are selected to create a packet length distribution. The packet lengths are then divided into 15 groups with a bin size of 100 bytes, representing packet lengths from 0 to 1500 bytes. The proposed method aims to classify and recognize DDoS attack signatures within a 15-second observation time interval, using a feature set of 15 features representing the packet length distribution of 5µs IAT packets. The number of packets is counted for each bin of each time interval, and a logarithm is used to round the number of packets to the level of packet count due to the large and varying range of packet counts. The resulting bin-1 and bin-2 packet length distribution of the CIC-DDoS 2019 training dataset for 5µs IAT packets in packet count level can be used as features for the proposed model.

The HTM model is capable of incremental learning in a single pass, allowing for the addition of new patterns without the need to retrain the model. This updating process is similar to the training phase, and both the Spatial Pooler and Temporal Memory are updated when a new pattern is introduced to the HTM model. Additionally, the HTM model supports an iteration mode, where patterns can be learned multiple times to improve detection performance. To further enhance the performance of the HTM model, supporting modules have been developed.

This model was evaluated on a dataset created by Combining the CICDDoS 2019 dataset and the MAWI dataset. The HTM-based continuous learning models were evaluated in two phases - before update and

after update. After updating the model, it achieved a perfect detection rate, accuracy, and precision for DDoS attacks such as LDAP, MSSQL, NetBIOS, UDP, and SYN. This shows that the HTM model was able to learn new patterns through continuous learning.

Metric	Attack types (Attack code)	Phase 1	Phase 2
Detection rate (%)	LDAP (2)	100	100
	MSSQL (3)	91	100
	NetBIOS (4)	96	100
	UDP (7)	36	100
	UDP-Lag (8)	0	68
	SYN (10)	78	100
Accuracy (%)	LDAP (2)	100	100
	MSSQL (3)	99	100
	NetBIOS (4)	99	100
	UDP (7)	95	100
	UDP-Lag (8)	96	98
	SYN (10)	99	100
Precision (%)	LDAP (2)	100	100
	MSSQL (3)	100	100
	NetBIOS (4)	96	100
	UDP (7)	34	100
	UDP-Lag (8)	0	84
	SYN (10)	100	100

Table 6: *Evaluation of HTM before and after update*

4.8. Challenges for future research

Although there are several ML algorithms in vogue in the realm of Cloud systems, Majority of the Research work has only been carried out in the areas of detecting DDoS by implementing mechanisms in Network layer by examining attack data with Classification models, whereas Self-triggered attacks and Application layer HTTP Flooding attacks need a whole different approach of detection. We need to detect more types of attacks across various other OSI layers [11].

Other challenges faced by these algorithms are:

- **Not Enough Label Data:**

There is not enough label data to train the ML model. This can result in poor accuracy in Detecting the DDoS attacks. We must always try to collect more data so that the model is trained to detect DDoS attacks by identifying as many features as possible corresponding to it.

- **Ever-changing Type of DDoS:**

Another challenge that these models face is the evolving nature of DDoS attacks. Attackers always come up with new ways for disrupting the services. So we cannot train a model and be done with it. We need to continuously update the model and possibly the training data to detect the newer types of attacks.

- **High dimensional feature space:**

The use of cloud computing produces a significant amount of data that includes numerous features. The CIC-DDoS2019 dataset, for example, contains 80 features for different types of DDoS attacks. However, having too many features can pose a challenge for Supervised Machine Learning algorithms as it is crucial to choose the most relevant features that provide the most distinctive information about DDoS attacks.

- **Class Imbalance:**

There is a lack of data available for various types of DDoS attacks. Most research has focused on the more prevalent attack types such as TCP SYN, UDP Flood, and IP Spoofing. Adequate data is necessary to train machine learning models to produce the most accurate classification results.

4.9. Results

The evolving area of using Machine learning to detect DDoS Attacks has seen a lot of research, and most of it has been conducted using datasets such as CAIDA, NSL-KDD, KDDCUP99, and CIC-DDoS19 for training and testing. Several supervised ML algorithms that were discussed above, upon experimentation, found to have achieved the accuracies in the range of 0.91 to 0.997, which is impressive. However, some studies have used more advanced deep learning models and achieved even better results. Research works such as Big Data and Deep Learning-based Approaches, and Hybrid Ensemble Learning models have achieved accuracy levels beyond 0.99. Therefore, it is becoming increasingly clear that deep learning is a promising approach for improving DDoS detection.

Using Multi Layer Perceptron (MLP) , which is a supervised ML technique that is optimized using ANN, upon experimentation it achieves an accuracy of 0.9863. Though MLP takes longer time to train, it is a promising technique which can be put into use effectively. Many novel types of DDoS attacks such as HTTP Flood, LDAP DDoS, smurf attacks have been effectively identified by Deep Learning Models whereas Supervised Machine Learning approaches have been nearly untested for these novel DDoS attacks.

Model under study	Dataset	Accuracy
SVM	NSL-KDD	0.997
Lightweight Deep Learning + GBDT	CIC-DDoS19	0.96
Perplexed-bayes classifier with feature selection	NSL-KDD & KDD-99	0.9915
MLP + ANN	Own Dataset	0.9863
Big Data and Deep Learning based Approach on Gateway Router	KDDCUP99	0.9973
CNN and DNN + Forward Feature Selection	CICIDS2017	0.9809
Deep Transfer Learning	Own Dataset	0.9928
Hierarchical Temporal Memory	CIC-DDoS 2019	0.996

Table 7: *Comparison of Algorithms with their dataset*

Although each machine learning algorithm has its own strengths and weaknesses, we found these to be the best performers. These algorithms have been successful in detecting DDoS attacks, and we suggest that they be considered for future applications in cloud systems.

5. Conclusions and Recommendations

With more and more organizations and applications migrating to cloud systems, the security challenges faced are also ever evolving. Also, attackers are changing and finding new ways to attack and compromise a system. This can lead to various outcomes from loss of critical information to loss of business and financial loss. Thus, it's imperative to conduct research in this particular field. Hence, in this report, we mainly focussed on one such class of cyberattacks: DDoS attacks.

Some of the recent devastating DDoS attacks include the Mirai botnet attacks in 2016, which targeted internet infrastructure and caused widespread outages, and the 2018 GitHub attack, which caused intermittent outages for several days. In 2020, one of the most extreme DDoS attacks occurred when Amazon Web Services was targeted, while cybersecurity researchers from Akamai discovered a botnet that is reportedly capable of launching 3.3 Tbps DDoS attacks. Additionally, in 2021, there were reports of DDoS attacks on the Belgian government, causing network outages and affecting public services.

Despite the lack of a uniform classification system, many researchers and organizations have developed their own taxonomies or frameworks for classifying DDoS attacks. These frameworks often categorize attacks based on factors such as the type of traffic used (e.g., ICMP, SYN, UDP), the duration of the attack, the number of sources involved, and the target system or network. By using a structured approach to classification, such as using attack vectors as the basis for DDoS taxonomy, researchers and organizations made huge strides to understand the nature of DDoS attacks and develop more effective strategies for preventing and mitigating them.

During literature review, we examined the conventional techniques used for detecting DDoS attacks and identified the reasons why they are inadequate for dealing with the current situation of vast amounts of data in the cloud that require DDoS detection. Although these methods are crucial for identifying known DDoS attacks, they can be easily inundated by the sheer number of requests, both valid and malicious, which can lead to system overload.

To prevent this, machine learning algorithms must be widely adopted by organizations for DDoS detection and mitigation due to their unparalleled ability to analyze large amounts of data and detect anomalies. This report provides evidence that machine learning algorithms are an effective tool for detecting DDoS attacks in cloud systems. An extensive literature review was conducted on various academic works that compare the effectiveness of traditional machine learning algorithms and specific hybrid machine learning models. The goal was to discover stronger defense strategies against DDoS attacks and to investigate opportunities for enhancing the detection of such attacks. This has significant

implications for cloud service providers, who could utilize these algorithms to enhance the security of their systems.

The biggest challenge is the availability of enough labeled data to train the machine learning algorithms. Moreover, the availability of high-quality labeled datasets is essential for training and evaluating machine learning models. Therefore, researchers should focus on creating and sharing datasets that represent various types of DDoS attacks and network environments. We recommend people to use the CIC-DDoS2019 dataset created by Canadian Institute for Cybersecurity (CIC) at the University of New Brunswick and is based on real-world DDoS attacks that occurred in 2019. It contains latest information on normal and malicious traffic which provide a good idea of different types of DDoS Attacks though it's also necessary to create newer datasets that are representative of the different types of DDoS Attacks.

Moving on to practical applications of machine learning algorithms for organizations, for any cloud system, traditional DDoS detection systems must be implemented in parallel with simple machine learning algorithms to detect DDoS attacks such as SVM, C4.5 and perplexed bayes classifiers. They provide good protection against various different types of DDoS attacks and are also cost-effective and easier to implement.

For organizations, the same is not applicable as they have a large quantity of network traffic and these algorithms cannot handle large amounts and dimensions of data. Organizations should invest more into combating this problem and implement deep learning solutions. We recommend creating groups of organizations that use the same cloud provider so that a FLAD model can be implemented among all of them while ensuring that no data is transferred or shared between the central server and also all the clients. These provide two major advantages to the organizations - same cloud providers can have similar defects and security concerns thus they can solve the problem at provider level and maintain the solution between the organizations thus reducing the cost. Also, they will be able to train the machine on various types of DDoS attack patterns and identities. The same attacker may attack the two systems in the same way. Through FLAD, the organizations can respond faster and better than through training the model on just their own data.

For a more secure cloud we recommend:

- Applying traditional methods such as an NSG that can only allow known traffic within the cloud server
- Applying Support Vector Machine model for detecting DDoS attacks at Gateway router which can read all the traffic and block malignant traffic.

- Regularly update the model based on the traffic received and also with new datasets which have been released.

Moving on to discussing the direction of future research, primarily we identify three main directions that warrant further exploration. They are as follows.

1. Waiting for the attack to reach the host devices of any cloud platform to detect DDoS attacks is not reasonable and we need to research and build defense mechanisms that detect attacks way before they reach the victim network. We need to conduct more research on hybrid approaches that detect such attacks on the edge of the network using various devices like IoT devices, IoT sensors, etc. We could perform neural network pruning so that low-powered devices can store and run the machine learning model. One could perform transfer learning so that it would greatly reduce the amount of time required to train and evaluate the machine learning model. [22] presents a two-stage edge detection by deploying ANNs at the consumer edge and LSTMs at the ISP's edge. More research is needed in terms of the machine learning algorithms tested and more hybrid edge detection architectures for cloud computing platform defense.
2. Source-side detection using machine learning is an emerging area of research in the field of DDoS attack detection. This approach involves using machine learning algorithms to identify the sources of malicious traffic in real-time, allowing network administrators to quickly block these sources and diminish the impact of an attack. While there have been some promising results in recent studies on source-side detection using machine learning, more research is needed to fully explore the potential of this approach. More focus needs to be given to research for detection from the source side as they can thwart attempts to compromise the cloud servers before the attack can even reach the target systems.
3. One potential avenue for future research, which is underrated and often overlooked, is to explore the motivations of those who initiate DDoS attacks as mentioned in [25]. This could involve the development and implementation of novel policies and strategies to counteract such attacks. By delving deeper into the mindset and incentives of attackers, researchers could uncover valuable insights that may inform more effective approaches to preventing and mitigating DDoS attacks in the future. Therefore, the study of attackers' motives represents a promising and worthwhile area of inquiry that could yield significant benefits in the ongoing battle against cyber threats.

6. References

[1] Zecheng He, Tianwei Zhang, Ruby B. Lee - Machine Learning Based DDoS Attack Detection From Source Side in Cloud, IEEE 4th International Conference

[2] J. Mirkovic, G. Prier, and P. Reiher, “Attacking ddos at the source,” in Network Protocols, 2002. Proceedings. 10th IEEE International Conference on. IEEE, 2002, pp. 312–321.

[3] Mirkovic, J., Prier, G., & Reiher, P.L. (2003). Source-end DDoS defense. Second IEEE International Symposium on Network Computing and Applications, 2003. NCA 2003., 171-178.

[4] T. M. Gil and M. Poletto, “Multops: A data-structure for bandwidth attack detection.” in USENIX Security Symposium, 2001, pp. 23–38.

[5] CS3, Inc. MANAnet Reverse Firewall. Available at: http://www.cs3-inc.com/MANAnet.html.

[6] Narendra Mishra, R. K. Singh, S. K. Yadav, "Detection of DDoS Vulnerability in Cloud Computing Using the Perplexed Bayes Classifier", Computational Intelligence and Neuroscience, vol. 2022, Article ID 9151847, 13 pages, 2022.

[7] Cohan Sujay Carlos. 2015. Perplexed Bayes Classifier. In Proceedings of the 12th International Conference on Natural Language Processing, pages 118–123, Trivandrum, India. NLP Association of India.

[8] M. Alkasassbeh, A. F. Sheta, H. Faris, and H. Turabieh, “Prediction of pm10 and tsp air pollution parameters using artificial neural network autoregressive, external input models: A case study in salt, jordan,” Middle-East Journal of Scientific Research, vol. 14, no. 7, pp. 999 – 1009, 2013.

[9] H. Faris, M. Alkasassbeh, and A. Rodan, “Artificial neural networks for surface ozone prediction: models and analysis,” Polish Journal of Environmental Studies, vol. 23, no. 2, 2014.

[10] O. Adwan, H. Faris, K. Jaradat, O. Harfoushi, and N. Ghatasheh, “Predicting customer churn in telecom industry using multilayer perceptron neural networks: Modeling and analysis,” Life Science Journal, vol. 11 , no. 3, pp. 75–81, 2014.

[11] Mouhammd Alkasassbeh, Ghazi Al-Naymat, Ahmad Hassanat, Mohammad Almseidin - Detecting Distributed Denial of Service Attacks Using Data Mining Techniques; (IJACSA) International Journal of Advanced Computer Science and Applications,Vol. 7, No. 1, 2016.

[12] B. Krse and P. van der Smagt, An Introduction to Neural Networks. CRC Press, The University of Amsterdam.

[13] L. V. Fausett, Fundamentals of Neural Networks: Architectures, Algorithms, and Applications. Prentice Hall, 1994.

[14] V. M. Patro and M. R. Patra, “Augmenting weighted average with confusion matrix to enhance classification accuracy,” Transactions on Machine Learning and Artificial Intelligence, vol. 2, no. 4, pp. 77 –91,2014.

[15] S. Sumathi, R. Rajesh and N. Karthikeyan - DDoS Attack Detection Using Hybrid Machine Learning Based IDS Models, Journal of Scientific & Industrial Research Vol. 81, March 2022, pp. 276-286.

[16] Muraleedharan N, Janet B, A flow-based anomaly detection system for slow DDoS attack on HTTP, CASB (2021) 29–45.

[17] K. P. Reddy, S. Kodati, M. Swetha, M. Parimala and S. Velliangiri, "A Hybrid Neural Network Architecture for Early Detection of DDOS attacks using Deep Learning Models," 2021 2nd International Conference on Smart Electronics and Communication (ICOSEC), Trichy, India, 2021, pp. 323-327, doi: 10.1109/ICOSEC51865.2021.9591969.

[18] J. He, Y. Tan, W. Guo and M. Xian, "A Small Sample DDoS Attack Detection Method Based on Deep Transfer Learning," 2020 International Conference on Computer Communication and Network Security (CCNS), Xi'an, China, 2020, pp. 47-50, doi: 10.1109/CCNS50731.2020.00019.

[19] B. McMahan, E. Moore, D. Ramage, S. Hampson, and B. A. y Arcas, “Communication-efficient learning of deep networks from decentralized data” ,2017, in Artificial intelligence and statistics, pp. 1273–1282.

[20] Roberto Doriguzzi-Corin, Domenico Siracusa,“FLAD: Adaptive Federated Learning for DDoS Attack Detection”, 2022, arXiv: 2205.06661v3

[21] Jema David Ndibwile; A. Govardhan; Kazuya Okada; Youki Kadobayashi, “Web Server Protection against Application Layer DDoS Attacks Using Machine Learning and Traffic Authentication”, 2015 IEEE 39th Annual International Computers, Software & Applications Conference, 0730-3157/15 © 2015 IEEE DOI 10.1109/COMPSAC.2015.240.

[22] Sowmya Myneni, Ankur Chowdhary, Dijiang Huang, Adel Alshamrani, SmartDefense: A distributed deep defense against DDoS attacks with edge computing, Computer Networks, Volume 209, 2022, 108874, ISSN 1389-1286.

[23] Mahjabin T, Xiao Y, Sun G, Jiang W. A survey of distributed denial-of-service attack, prevention, and mitigation techniques. International Journal of Distributed Sensor Networks. 2017;13(12). doi:10.1177/1550147717741463

[24] Christos Douligeris, Aikaterini Mitrokotsa, “DDoS attacks and defense mechanisms: classification and state-of-the-art”, Computer Networks, Volume 44, Issue 5, 2004, Pages 643-666, ISSN 1389-1286,

[25] S. T. Zargar, J. Joshi and D. Tipper, "A Survey of Defense Mechanisms Against Distributed Denial of Service (DDoS) Flooding Attacks," in IEEE Communications Surveys & Tutorials, vol. 15, no. 4, pp. 2046-2069, Fourth Quarter 2013, doi: 10.1109/SURV.2013.031413.00127.

[26] L. S. Matsa, P. G. -A. Zodi-Lusilao and P. F. Bhunu-Shava, "Forward Feature Selection for DDoS Detection on Cross-Plane of Software Defined Network Using Hybrid Deep Learning," 2021 3rd

International Multidisciplinary Information Technology and Engineering Conference (IMITEC), Windhoek, Namibia, 2021, pp. 1-7, doi: 10.1109/IMITEC52926.2021.9714561.

[27] Gupta, B.B., Badve, O.P. Taxonomy of DoS and DDoS attacks and desirable defense mechanism in a Cloud computing environment. Neural Comput & Applic 28, 3655–3682 (2017).

[28] Hussain, A., Heidemann, J., and Papadopoulos, C. 2003. A framework for classifying denial of service attacks. In Proceedings of the ACM SIGCOMM Conference (Karlsruhe, Germany). 99--110.

[29] J. Mirkovic and P. Reiher, A taxonomy of DDoS attack and DDoS defense mechanisms, ACM SIGCOMM Computer Communications Review, vol. 34, no. 2, pp. 39-53, April 2004.

[30] T. Peng, C. Leckie, and K. Ramamohanarao, Survey of network-based defense mechanisms countering the DoS and DDoS problems, ACM Comput. Surv. 39, 1, Article 3, April 2007.

[31] RioRey, Inc. 2009-2012, RioRey Taxonomy of DDoS Attacks, RioRey Taxonomy Rev 2.3 2012, 2012.

[32] S. Ranjan, R. Swaminathan, M. Uysal, and E. Knightly, DDoS-Resilient Scheduling to Counter Application Layer Attacks under Imperfect Detection, IEEE INFOCOM'06, 2006.

[33] Arbor Application Brief: The Growing Threat of Application-Layer DDoS Attacks, Arbor Networks, Feb. 28, 2011, [online].

[34] Darwish, M., Ouda, A., & Capretz, L. F. (2013, June). Cloud-based DDoS attacks and defenses. In International Conference on Information Society (i-Society 2013) (pp. 67-71). IEEE.

[35] DDoS Attack Detection using Machine Learning Techniques in Cloud Computing Environments Marwane Zekri, Said El Kafhali, Noureddine Aboutabit and Youssef Saadi, 2017 3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)

[36] DDoS Attack Detection Using Hybrid Machine Learning Based IDS Models S Sumathi, R Rajesh and N Karthikeyan, Journal of Scientific & Industrial Research Vol. 81, March 2022, pp. 276-286

[37] B. B. Gupta, A. Gaurav and D. Peraković, "A Big Data and Deep Learning based Approach for DDoS Detection in Cloud Computing Environment," 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE), Kyoto, Japan, 2021, pp. 287-290, doi: 10.1109/GCCE53005.2021.9622091.

[38] A. R. Wani, Q. P. Rana, U. Saxena and N. Pandey, Analysis and Detection of DDoS Attacks on Cloud Computing Environment using Machine Learning Techniques

[39] Makkawi, Ahmed Mohammed, and Adil Yousif. "Machine Learning for Cloud DDoS Attack Detection: A Systematic Review." 2020 International Conference on Computer, Control, Electrical, and Electronics Engineering (ICCCEEE). IEEE, 2020.

[40] B. B. Gupta, A. Gaurav and D. Peraković, "A Big Data and Deep Learning based Approach for DDoS Detection in Cloud Computing Environment," 2021 IEEE 10th Global Conference on Consumer Electronics (GCCE), Kyoto, Japan, 2021, pp. 287-290, doi: 10.1109/GCCE53005.2021.9622091.

[41] M. -H. Nguyen and Y. -K. Lai, "Implement a Continuous Learning Model to Detect Different Types of DDoS Attacks with Hierarchical Temporal Memory," 2022 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC), Chiang Mai, Thailand, 2022, pp. 1780-1785, doi: 10.23919/APSIPAASC55919.2022.9980114.