Intelligent Firewall Automation For Virtualized Cloud Infrastructure

MSc Research Project

Cyber Security

Shivani Arya

Student ID: x23195304

School of Computing
National College of Ireland

Supervisor: Rohit Verma

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Shivani Arya		
Student ID:	x23195304		
Programme:	Masters in Cybersecurity	**Year:**	2024-25
Module:	Research Project		
Lecturer:	Rohit Verma		
Submission Due Date:	12-12-2024		
Project Title:	Intelligent Firewall Automation For Virtualized Cloud Infrastructure		
Word Count:	7788 words	**Page Count:**	22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: ………

Date: ………

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Intelligent Firewall Automation For Virtualized Cloud Infrastructure

Shivani Arya
x23195304

Abstract

Cloud computing is a relatively young innovative solution that has been progressively implemented by many enterprises around the world; however, its security risks are rather high, and one of the most critical is DDoS. It is very sad that these attacks can hugely affect organizations depending on cloud services in their operations. This work discusses DDoS mitigation measures on AWS platform while identifying the cloud solutions provider serving over 1.7 million active customers in more than 190 countries. The work assesses the safeguards that AWS provides to mitigate and prevent DDoS attacks regarding the firm's infrastructure. Live experiments consisting of, at least, tens of concurrent connections originating from different AWS instances were carried out in order to test the ability of instances to withstand simulated DDoS threats resulting from such scripts. ML methods, viz. Decision Trees, Random Forest, and Sector Vector Machine became part of the system to prevent malware attacks in real time. The implementation relies on AWS EC2 instances, as the basis for applying intelligent firewall and hosting of important web applications. This work offers knowledge on how to strengthen security criteria while utilizing automation and ML to detect anomalies in AWS-based systems and prevent emerging threats.

1 Introduction

In the computing environment, cyber threats have changed in terms of intensity and pervasiveness, with significant negative impacts on various organizations irrespective of their size or industry. Cybercriminals have over the recent past stepped up their operations and this started about a decade ago and was directed to both big firms and small ones. As stated by Femminella and Reali *et al*. (2024), Distributed Denial of Service & Denial of Service are some of the most familiar types of cyber threats. A DoS attack generally involves raising the volume of traffic that passes through a server or computer network to a level whereby the target is unable to respond to requests while still working. More advanced technical methods and more frequent appearance of DDoS attacks have been observed recently, which indicates the further significance of developing effective security strategies for defending against these threats.

Cloud computing, which is the powerhouse of today's business solutions, is the provision of scalable and affordable services that allow data to be stored and applications to run from a cloud. AWS is one of the industry leaders in this regard providing several cloud-based solutions, including computing, storage, and networking (Chimuco *et al*. 2023). AWS has described its services as a pay-as-you-go model and virtual deployment that have made companies around the world prefer AWS. Through compute instances, storage choices, and database services it offers businesses a flexible environment through which they can host complex applications at low or no initial capital investment and operating expenses.

AWS has a service called Web Application Firewall (WAF) that will shield applications from common web exploits that may impact performance, security, and availability. The WAF assists in filtering and analysing HTTP and HTTPS connections and comes equipped allowing it to be adjusted to the application's requirements (Chang *et al*. 2022). However, WAF has its shortcomings in many cases because it cannot prevent certain types of attacks, such as DDoS, as in limitation cases, it mainly utilizes patterns and configurations as important vectors. For this purpose, the highly evolved security measures, like the usage of the firewalls integrated with the algorithms of Machine learning are under consideration to bring superior security to the system.

1.1 Artificial Intelligence in Cybersecurity

Artificial Intelligence is among the most disruptive technologies that are adopted in the cybersecurity landscape especially, in creating the improved firewalls called the Next Generation Firewalls (NGFW). Conventional firewalls operate on a set of rules installed on the appliance and block traffic based on these rules; they do not learn, and they do not adjust their mode of operation based on current or real-time data (Cunha *et al*. 2024). AI firewalls are digital systems that analyse the traffic in real time, recognize the threats, and change their corresponding configurations to counter threats automatically. This cooperation in utilizing AI in cybersecurity implies that the defence system is alive with the ability to learn about new threats and defend against them more proactively than through interventions that are automated.

Modern intelligent firewalls can process huge quality of data compared to network traffic and recognize patterns and outliers that could mean that an attack is underway. For example, Random Forests, Decision Trees, and Support Vector Machines (SVM are some of the algorithms widely employed in such cases to categorize traffic as either legitimate or malicious. These are systems that are modelled to analyse information on previous attacks and then adapt to identifying new and emergent threats (Pandey *et al*. 2023). By applying machine learning algorithms, it will be possible to achieve maximum protection against malware, phishing attacks, and APTs, which are hard to contain using typical security measures.

1.2 Cybersecurity Issues in Cloud Computing

There are numerous advantages of cloud computing although there are many issues of cybersecurity that it brings along. Cloud environments and their services are recognized as active and quickly changing because resources can be easily launched and shut down (Chughtai *et al*. 2024). Cloud infrastructures are distributed across different geographic regions and entail a network that is even more vulnerable to cyber threats.

This paper identifies one of the primary threats to the sharing of infrastructure in cloud environments: namely, the protection of hosted data and applications against threats. In the multitenant-cloud environment, different organizations all use the same physical infrastructure which may increase the risk of data leakage when security protocols are not enforced (Falkner and Apostolopoulos *et al* 2022). For instance, with DDoS attacks, the attacker focuses on the availability of cloud services, and therefore the cloud service provider must provide a strong countermeasure against the attacks to avoid bringing services down.

In 2024, worldwide organizations reported that phishing continued to be the most common security breach, occurring in both cloud and non-cloud environments, with 73% and 74% (Figure 1) of respondents indicating that their organizations faced this type of cyberattack. Moreover, the proportion of survey participants who experienced user account compromise incidents in the cloud was 38 percent, an increase from the 34 percent reported for on-premises cases.

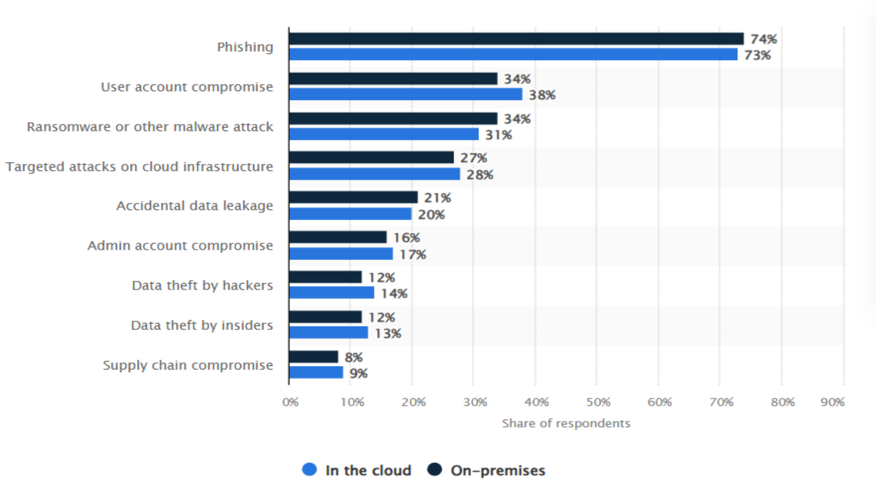


Figure 1: Most frequently occurring security incidents in the cloud and on-premises globally in 2024

1.3 Machine learning in cloud

Currently, many researchers have demonstrated their concern in analysing these attacks based on various approach (Kushwah Gopal Singh *et al*. (2022).In the security system driven by machine learning, unusual and normal behaviours are classified under various labels through training models. Using machine learning, different types of features are obtained, which aid in the detection of cloud attacks. Although these techniques identify the various attack types over the rich cloud environment, still they lack the ability to cater for unknown attacks.

1.4 Research question

- To what extent the machine learning algorithms namely Decision Trees, Random Forest and Support Vector Machine improves identification and reduction of DDoS attacks?
- How can the intelligent firewalls enhance the CPU efficiency and regulate the network traffic while keeping the performance to optimal during attack scenarios?

1.5 Objective and motivation

The primary goal is to optimize the security of the external layer of the application in AWS that contains an intelligent firewall. The study specifically seeks to address three objectives as mention below:

- Automating traffic filtering through machine learning approach
- Continuously control and manage CPU and network resources.
- To analyse the effectiveness of Decision Tree, Random Forest and Sector Vector Machine in the detection and prevention of DDoS attack with respect to the degree of accuracy, time, and precision.

Consequently, this research aims at proposing an intelligent firewall system which implements AI principally to safeguard cloud-based Virtualized environments from outer threats with a focus on DDoS attacks. Using the Decision Trees, Random Forests, and SVM algorithms the study proposes to improve the firewall's detection capabilities as well as the ability to find a balance for CPU and network traffic usage. The study will deploy the implementation of AWS cloud tools, such as EC2 instances, and CloudWatch monitoring tools to prove the feasibility and applicability of this approach. This research also aims to fill the voids left open by current security tools or services, including adaptability in the physical infrastructure, automation and resource management, to offer strategic references for the succeeding cybersecurity framework.

The topic of cloud attacks detection has always been an interesting one, and it remains so until the present day. In fact, there are many research papers where authors discussed the subject and proposed their solutions to this problem. It became known that several studies used a various kind of methods, including keep learning, machine learning artificial intelligent and several more to approach the issue.

2 Related Work

The identification of cloud attacks has always been an interesting area, and it still is. In fact, authors have taken their time and given solutions to this problem in various journal articles. Therefore, multiple papers applied different methods categories like deep learning, machine learning, artificial intelligent and multiple others to address this problem.

2.1 Securing AWS cloud from Vulnerabilities

One of the areas of focus in the study by Waseem *et al*. (2017) is that the Distributed Denial-of-Service (DDoS) attack may impact AWS computing. The study focuses on the growth of using Elastic Load Balancer, Distribute Traffic, Overload and AWS Web Application Firewall for the application layer, Using AWS DDOS best practices, they explain how to focus on network devices isolated and monitoring .The impact that DDOS assaults experience on cloud service availability and organization resilience is not investigated in this work, Furthermore the machine learning methods for real-time cyber-security detection and prevention are not being integrated to enhance the ability of such systems to confirm by time-based suggestions.

Advantages: According to the proposed solutions, there are enhancements in the security protection implemented by AWS features hence improving security for applications and data.

Disadvantages: Removing all three of these factors, scaling up comes with high costs, introducing a potential flood of false positives that will be detrimental to the user experience, and relying solely on AWS.

Future Scope: Future work may consider directions extending threat identification solutions based on artificial intelligence and machine learning for DDoS mitigation in real time.

2.2 The Role of Next-Generation Firewalls

The research done by Himanshu *et al.* (2021) focuses on the effectiveness of artificial intelligence approach on the firewall to enhance security. Different methods and algorithms like Recurrent Neural Networks and Convolutional Neural Networks are analyzed with reference to real time threat detection and prevention. These models increase threat detection efficiency while the decision making in these algorithms may not be well understood. Many efforts have been made to improve the applicability for deep learning approach, but there has been less focus on making them more understandable, especially about what kind of mistakes the model makes during false positives.

Advanced Threat Recognition for AI firewalls is aimed at using sophisticated mathematical models for identifying and countering new- generation threats in a shorter time, thus strengthening overall security. By applying these advanced features in a filtering system, it is possible to distinguish between a true attack and other types of traffic, which in turn leads to a more efficient system and less false positives.

Disadvantages: Another main disadvantage of the deep learning algorithm is that how decision are made can be obscured, and, therefore, it is challenging to know how specifically threat is recognized. Predicting continues to be a big issue in many AI algorithms, especially in the aspects of the volumes and complexities of network traffics that may cause a compromise on the performance desired.

Future Scope: It would be crucial to carry out further research in order to discover methods that will help to improve the interpretability of AI models to gain public's confidence in their results. The future work must consider dynamic retraining and scalability issues to meet the requirements of AI-based firewalls as new types of threats appear in the future.

On similar grounds, Ahmadi *et al.* (2023) mentioned area of interest happens to be next-generation AI-based firewalls. Although, these firewalls have shown high detection accuracy, they are plagued with problems of interpreting deep learning models. There is an important lack of information as to how deep learning models perform their decision-making, an aspect that is crucial needed in order to combat problems like false positives and improve of the reliability of such systems.

2.3 Analysis of Firewall Configuration Automation in Virtualized Infrastructure

Daniele *et al.* (2022) have studied the problem of automating the process of configuring firewalls *in* virtual networks to solve the problem of manual configuration that causes many security threats and inefficiencies. The goal is to provide the tool that goes beyond simple automatic generation and deployment of firewall rules, and also to provide the method of proving the correctness and optimality of the generated configuration. Thus, implementing all these elements, the research aims at improving the security situation in the virtualized environment and refining the possibilities of the further network security development according to the tendencies in threats' appearance and expressions, as well as the tendencies in the demand's evolvement.

It must be noted that while the current methodology is based on packet filters there is possibility to expand the approaches further to involve such other NSFs Network Functions Virtualization as web application firewalls, anti-spam filters as well as intrusion detection systems. This would enable the development of a broader security paradigm capable of accommodating other types of threat and demand.

Advantages: It is efficient because it means that the original firewall is set up automating butting and competitiveness, which is many times faster that manually configuring a firewall. Through minimizing the possibility of human inaccuracy in security methods, it enables the development of more effective and more successful ones. They also found that the software helped network administrators avoid wasting time on repetitive configuration work so serve to increase efficiency.

Disadvantages: While this increases the efficiency of the solutions, it makes it challenging to deliver complex configurations based on the peculiarities of a particular or special type of security need, and may result in overlooking these requirements. Automation is wholly dependent on the configuration & parameters formulated at the start of the automation process; improper configuration can lead to insecure or even sub-optimal performance.

Future Scope: Further in this study more research can be made to decide the Firewall configurations with the help of AI and machine learning in order to respond quickly to new threats. Innovative approaches to Firewall that would conform more systems and network architectures would enable wider deployment of automated Firewall settings.

2.4 Techniques of Machine Learning for Enhancing Cloud-Based Cybersecurity

The study conducted by Rexha et al. (2023) examines machine learning algorithms that are used to detect Distributed Denial of Service (DDoS) threats. The Authors have discussed the below mentioned algorithms:

Decision Trees: This algorithm is described as one of the most successful in the analysis as it has a high accuracy, precision, and recall for DDoS attacks identification.

Support Vector Machine (SVM): SVM is applied for grouped and its performance in the aspect of detecting attacks is compared against other algorithms.

Naive Bayes: This probabilistic classifier is also integrated in the evaluation in order to determine its ability to classify traffic as either legitimate or malicious.

K-Nearest Neighbors (KNN): KNN is also included in the comparative assessment of other algorithms for DDoS detection

The current detection techniques seem like they shall lose efficiency as the many threats develop. What is needed, are methods that can be learned, allowing the identification and detection of new attack paradigms. According to the paper, there is an opportunity to use a mixture of various machine learning approaches to improve detection performance. However, the basic idea of using only one type of algorithm has not yet been studied, while fewer studies focus on the hybrid models that combines the advantages of different algorithms. The results shown here suggest that the Decision Trees algorithm ranked the highest compared with the others in the test case study showing potential in monitoring DDoS attacks in a cloud environment with greater accuracy and efficiency.

Sherwin *et al*. (2022) in their research discussed the effects of DDoS attacks on the cloud computing system, and to discuss the different machine learning methods useful in the detection of these attacks. This research is intended to assess the drawbacks of established systems and pave the way for a robust framework reflecting suitable technologies for effective machine learning that ensures the safeguard of cloud services from DDoS threats. While there are studies already in machine learning techniques for DDoS, the problem lies in the algorithm skill of adapt to new variants of these attacks. It also shows low accuracy against unseen attack patterns, this means existing models could be easily attacked by new attack patterns, this call for more research on more dynamic and adaptive learning algorithm

2.5 Application Layer DDoS Detection in SDN Environments

The application layer DDoS (AL-DDoS) attacks present critical problems in the network security domain because the attacks replicate the legal flow to avoid detection. However, autonomic attacks of this kind are more frequent at the present, and while research has been rather limited, most of it focused on threats at the network layer, the problem of attacks at the application layer remains uncovered (Kaur *et al.,* 2023). It also shows that there is little extant work targeted solely at application layer threats and how they can be detected. There is weakness in the systematic reviews of previous detection techniques specifically detection effectiveness indices including accuracy, computational cost, and practical applicability. The software-defined networking (SDN) architecture itself is also vulnerable to DDoS attacks considering regulations have the capacity to modifications. This double-barreled problem makes detection and remedy more challenging for researchers since they must tackle the attacks themselves and vulnerabilities of the network infrastructure. In the future, enhancement of DDoS application layer detection algorithms using machine learning and artificial intelligence could also be future research directions Possible future research areas include investigating the specific weaknesses introduced by software-defined networking (SDN) domain.

3 Research Methodology

This paper aims to propose and assess an intelligent firewall framework that safeguards infrastructures on AWS, especially against DDoS attacks. It is underpinned by survey and experimentation, evaluating the firewall's ability to address cyber threats while optimistically performing.

3.1 DDoS attack Architecture

Distributed Denial of Service attack is usually performed with many systems attacking one target. The hosts used in the target are normally under the control of a remote attacker who uses it to perform the attacks. In many instances, the resource owner is often unaware of the misdeeds committed via their system by the remote attacker.

There is a simulation of a DDoS attack from a Kali Linux machine through the use of MHDDoS tool. The attacks deployed are the billing website through HTTP GET floods, SYN floods as well as ICMP floods attacks. This realistic simulation produces high numbers of visitors to observe how the firewall prevents unauthorized access while permitting service access. These attacks are made to bring certain levels of intensity and long duration to test the ability of the system on different scenarios. A pictorial overview of above is given in Figure 2.

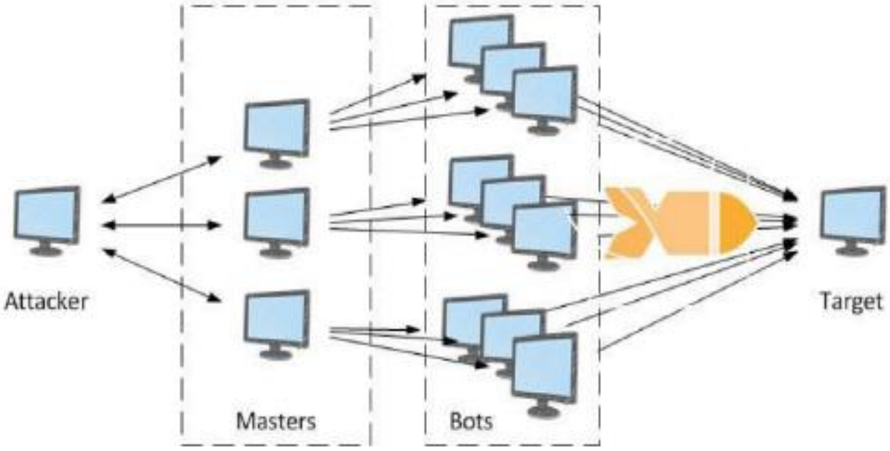


Figure 2: Overview of DDoS attack Architecture

3.2 Infrastructure Setup

The setup of test environment involves creation of test environment in the AWS cloud. For this purpose, two EC2 instances are launched First, Amazon Linux Operating System will be launched in an Elastic Compute Cloud (EC2) instance named as internal firewall. Next, the firewalled service at the firewall EC2 instance will be instantiated with specific rules for the IP filtering and rate limiting of the hostile traffic. NGINX will act as a reverse proxy and redirect the traffic to the billing website, and any malicious traffic will be logged and investigated later. Elastic Compute Cloud has a billing website where a WordPress application is placed in a public subnet within a different (VPC). After filtering out such threats, the firewall instance will lead all external traffic to the billing website. Backend systems and public resources are set up in their own security groups, routing tables or subnets. Elastic IP will be allocated for internet connectivity.

3.3 Machine Learning Models

Traffic data is utilized for building algorithms for machine leaning include Decision Trees, Random Forest, and SVM. As stated by Pandey *et al.* (2023), these models put traffic into one of two categories: benign or malicious, and their effectiveness is measured by detection rates, false positives and negatives, and computational cost.

Decision Trees: This algorithm provides an understandable and clear way for classifying network traffic. The decision-making process is made less complicated by dividing it into a tree structure where, through routine and set paper patterns, it is easy to define the traffic as either malicious or legitimate.

Random Forests: Random Forests have the ability to openly determine various traffic systems, anticipate traffic density, and even coordinate traffic movements in actual traffic situations. For this reason, they are well suited for the task at hand because they can process big datasets with many features.

In and out-line, Random Forests combine the advantages of other decision trees to provide a very reliable model most especially in places where the accuracy of the estimation is key.

Support Vector Machines (SVM): SVM is used because this approach works best when it is handling high-dimensional data. SVM performs optimization that constructs a hyperplane between normal and anomalous traffic and it does well in detecting neutrally anomalous traffic, especially on complicated data sets.

3.4 Data Collection

Metrics are pulled from AWS CloudWatch as well as logs from running instances of EC2, including network bandwidth and memory usage, and CPU performance indicators (Chang *et al*. 2022).

Phases of data collection

- **Pre-attack Phase:** AWS CloudWatch logs and metrics establish the baseline resource consumption of resources, including CPU and network traffic.
- **Attack Phase:** In a real DDoS attack, the system displays the details of the incoming traffic, blocked, and allowed requests, CPU spikes, and other network anomalies.
- **Post-Attack Phase:** Logs are analyzed to determine the recovery duration, resource stabilization, and efficiency of machine learning models in relieving the threats that still persist.

3.5 Evaluation Metrics

Measures of success are execution time, network bandwidth requirements, response time, and detection rates. These metrics are used in making comparisons between the firewall performance in normal situations and under attack (Cunha *et al*. 2024). The two most important design metrics about the firewall system are:

1. **CPU Utilization:** Computation load taken by the firewall in normal and attack scenarios, respectively, to know the utilization of the resources efficiently.
2. **Network Traffic:** The number of connections and demands the system can simultaneously accommodate in conditions of low latency and high incoming traffic, during the attack.

4 Design Specification

In this section we will discuss the flow of our proposed work for Cloud Infrastructures. The architecture implements traffic filtering, traffic monitoring and Machine learning Algorithm for Security of Web applications hosted in AWS cloud environment.

4.1 AWS Cloud Infrastructure Design

AWS Cloud Infrastructure Design provides the primary underlying architecture for the intelligent firewall system to provide security and performance. Creating a Website Virtual Private Cloud with a CIDR block of 13.0.0.0/16 gives us a logically separated net with full control over the network settings. There are two subnets one private where the backend information is located and one that is visible from the outside world. Such a network topology enables stronger isolation, and a lesser attack surface, and allows much more fine-grained control of traffic.

4.2 Firewall Configuration

The Internet Firewall EC2 Instance remains a crucial element of this design and provides the first line of defence against any such traffic. An overview of the cloud infrastructure is provided in Figure 3. This presented virtual machine uses the Amazon Linux operating system and Firewalld service, which appears to be a more flexible system in control of incoming traffic streams. Such a powerful configuration supports dynamic programmable traffic filtering mechanisms that can be adapted based on future cyber threats real time (Ahmadi and S, 2023). It uses mechanisms of rate control and IP filtering to distinguish between the patterns of the Distributed Denial of service (DDoS) and allow other valid requests. For Incoming traffic ports http (port 80), ssh (port 22) and https (port 443)) are open, rest all are closed. These measures make it possible for the system to continue resisting high traffic loads while continuing to provide access for legitimate users.

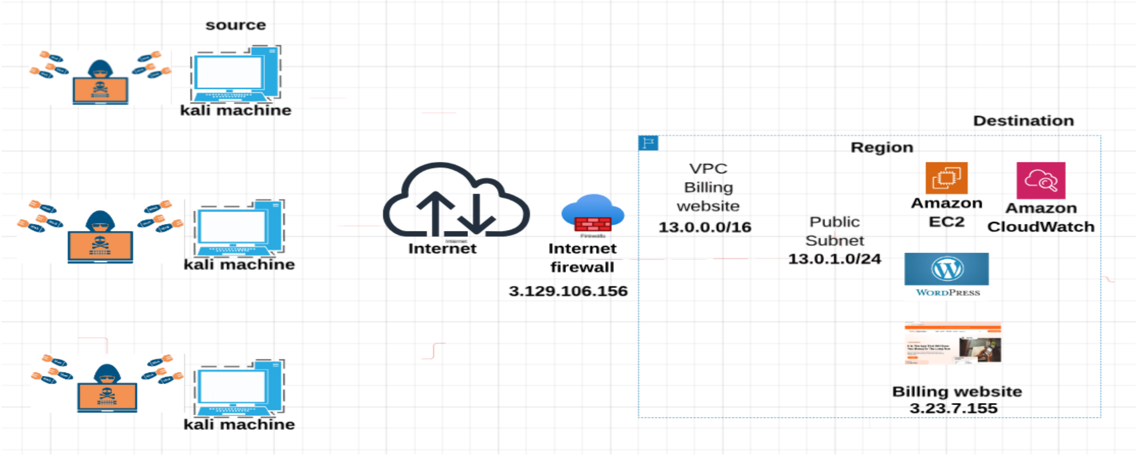
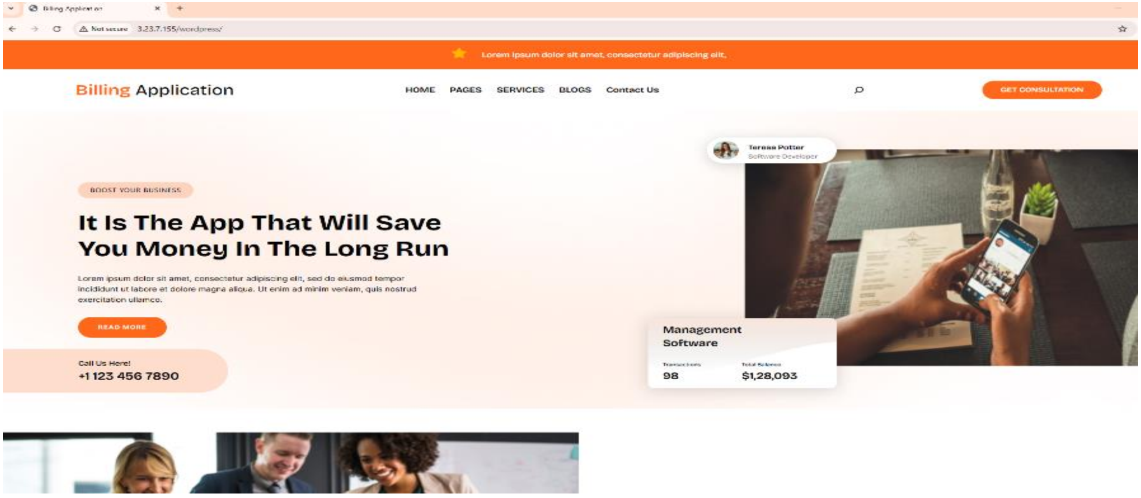


Figure 3: Overview of Cloud Infrastructure

4.3 Application Deployment

The target application, the Billing Website (Figure 4), is hosted on a separate EC2 instance situated in the Billing VPC's public subnet. This instance currently has WordPress installed; a content management system used in serving some of the most visited websites in the world in its actual operating environment. The backend is a secured MySQL database with fine-tuned credential permission. A specific security group serves as an essential control point by allowing traffic only from the IP of the intelligent firewall. The entire security of the application is also sustained by detailed file permissions settings and role-based access controls.



It Is The App That Will Save You Money In The Long Run

Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco.

Figure 4: The billing website that needs to be attacked

4.4 Monitoring and Alerting

AWS CloudWatch service has been configured to provide constant surveillance of key performance parameters of the firewall and the billing site instances. This comprises monitoring CPU usage and networking rates as depicted in Figure 5. Specific CloudWatch metrics have been set to set alarms depending on conditions such as heavy web traffic or consumption rates for any resource. This particular arrangement of monitoring helps the system in attending to any probable troubles that may affect the performance as well as availability of the applications.

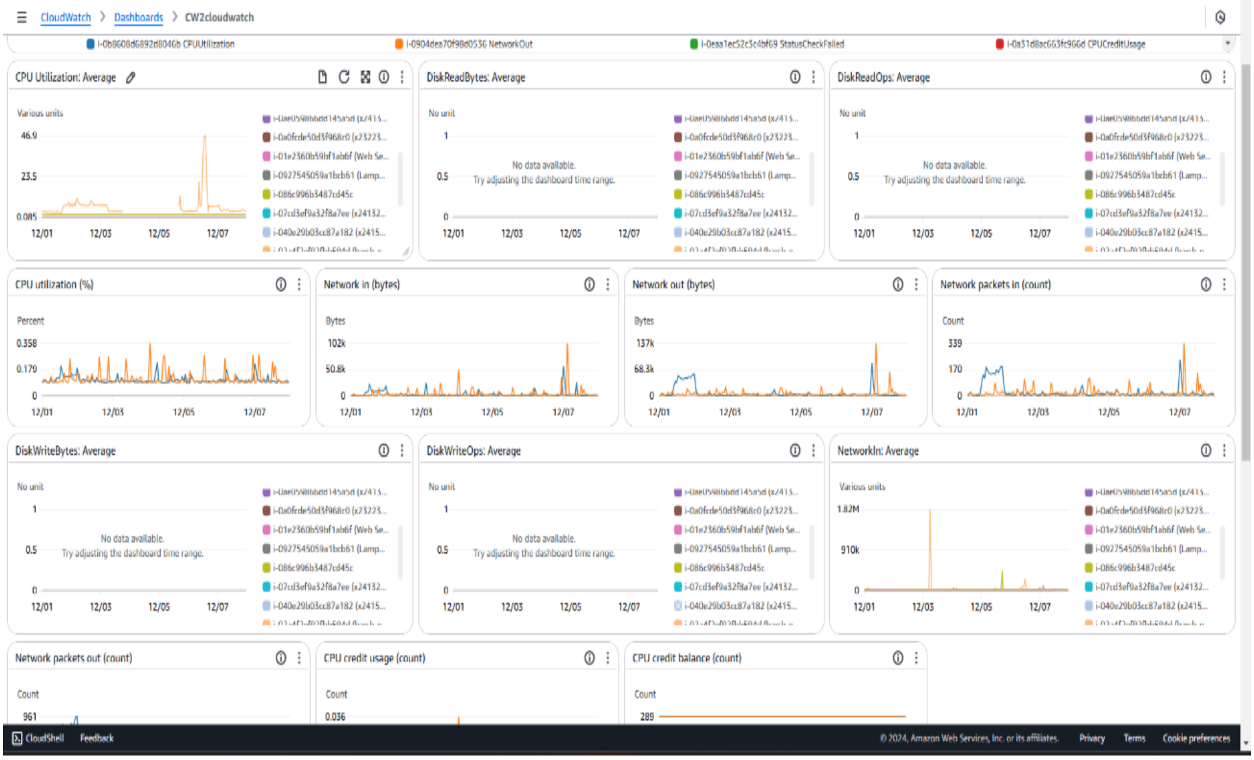


Figure 5: AWS CloudWatch Dashboard

5 Implementation

In the following section, we have proposed the intelligent firewall solution implementation based on open-source technologies, AWS services and a dedicated Python script. This method takes advantage of the features of an open-source software to develop an advanced firewall system, based on the Amazon Web Service that is scalable and dependable while adapting to an always changing security environment. It can be configured to change its behaviour with regards to certain events as well as make changes in network security more generally by having the firewall write Python scripts for the system. The overall goal of the complete solution is to offer optimal considerations of security and control within logical functions and low resources.

5.1 Tool and languages

The deployment and evaluation are provided by a combination of open-source tools and AWS services:

- **AWS Services:** The AWS services used in this system are Elastic Compute Cloud, Virtual Private Cloud, CloudWatch & Route Tables. All these are used for monitoring the infrastructure system.
- **Simulation Tool:** MHDDoS is a piece of software that simulates DDoS attacks and generate malicious traffic.
- **Python Libraries:** This system makes use of the Python libraries Scikit and NumPy for machine learning applications. Also, Matplotlib is used for graphical representation of data and psutil is used for the monitoring of computer resources. Taken together, these strong frameworks help in recording the right data and processing and visualizing them for understanding and analysis.

5.2 Attack Configuration

It was launched from an Oracle VM VirtualBox hosted system and a Kali Linux distribution. The firewall instance was used in emulating DDoS attacks on the billing site with the MHDDoS tool.

5.3 Execution of the Attack

For launching attack command are mentioned below: -

python3 start.py get http://3.129.106.156 1 1000 proxylist.txt 10 200 debug (Figure 6)

The name of the DDoS attack script to execute is ‘start.py’ The ‘start.py’ should be in the same directory with the script or in a directory well identified by the system.

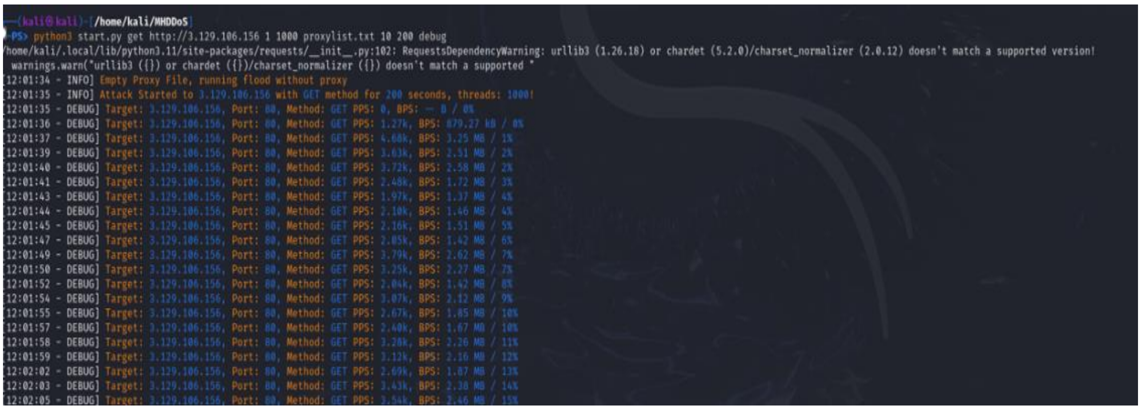
The script employs HTTP GET DoS attack method which encompass sending an immense number of GET requests to the intended website or IP address with the intention of freezing or overloading the target with the intention of freezing or overloading it.

The script can be configured with the following parameters:

- 1000 (Number of Threads): This specifies the number of threads at once that will mimic the HTTP requests so as to produce a more distributed traffic.

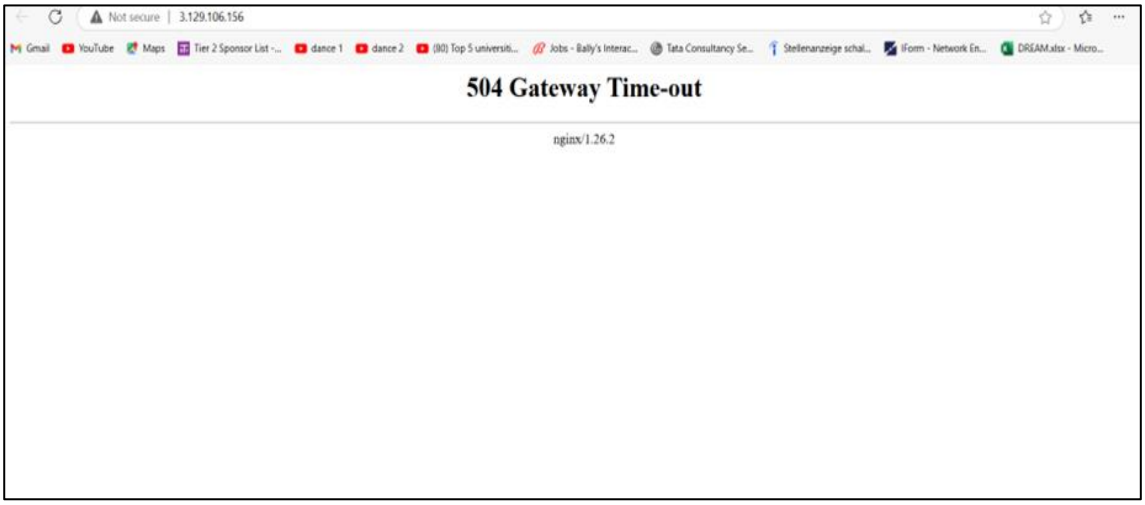
- proxylist.txt (Proxy List File): If proxies are going to be set then this file should contain proxy address on one line at least.
- 10 (Requests per Connection): It will make ten requests at a time and move on to another connection, and that, too, ten requests, and move to the next one, that is, to help keep the connections active and put more load on the target server.
- 1000 (Duration of the Attack): The attack will take 1000 seconds to execute and will pause automatically after that ensuring that the total time of attack does not exceed 16 minutes and 40 seconds. This duration can be as short or long as is necessary to meet the demands of relevant legislation, the organization, and customers.
- debug (Optional Debug Mode): If the debug mode is enabled, then more log information will be provided to control and investigate the advancement of the attack.

The website goes down after the attack is successful as shown in Figure 7.



```
┌──(kali㉿kali)-[/home/kali/MHDDoS]
└─PS> python3 start.py get http://3.129.106.156 1 1000 proxylist.txt 10 200 debug
/home/kali/.local/lib/python3.11/site-packages/requests/__init__.py:102: RequestsDependencyWarning: urllib3 (1.26.18) or chardet (5.2.0)/charset_normalizer (2.0.12) doesn't match a supported version!
  warnings.warn("urllib3 ({}) or chardet ({})/charset_normalizer ({}) doesn't match a supported "
[12:01:34 - INFO] Empty Proxy File, running flood without proxy
[12:01:35 - INFO] Attack Started to 3.129.106.156 with GET method for 200 seconds, threads: 1000!
[12:01:35 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 0, BPS: -- B / 0%
[12:01:36 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 1.27k, BPS: 879.27 kB / 0%
[12:01:37 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 4.68k, BPS: 3.25 MB / 1%
[12:01:39 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 3.63k, BPS: 2.51 MB / 2%
[12:01:40 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 3.72k, BPS: 2.58 MB / 2%
[12:01:41 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 2.48k, BPS: 1.72 MB / 3%
[12:01:43 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 1.97k, BPS: 1.37 MB / 4%
[12:01:44 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 2.10k, BPS: 1.46 MB / 4%
[12:01:45 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 2.16k, BPS: 1.51 MB / 5%
[12:01:47 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 2.05k, BPS: 1.42 MB / 6%
[12:01:49 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 3.79k, BPS: 2.62 MB / 7%
[12:01:50 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 3.25k, BPS: 2.27 MB / 7%
[12:01:52 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 2.04k, BPS: 1.42 MB / 8%
[12:01:54 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 3.07k, BPS: 2.12 MB / 9%
[12:01:55 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 2.67k, BPS: 1.85 MB / 10%
[12:01:57 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 2.40k, BPS: 1.67 MB / 10%
[12:01:58 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 3.28k, BPS: 2.26 MB / 11%
[12:01:59 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 3.12k, BPS: 2.16 MB / 12%
[12:02:02 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 2.69k, BPS: 1.87 MB / 13%
[12:02:03 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 3.43k, BPS: 2.38 MB / 14%
[12:02:05 - DEBUG] Target: 3.129.106.156, Port: 80, Method: GET PPS: 3.54k, BPS: 2.46 MB / 15%
```

Figure 6: The launching of DDoS Attack



504 Gateway Time-out

nginx/1.26.2

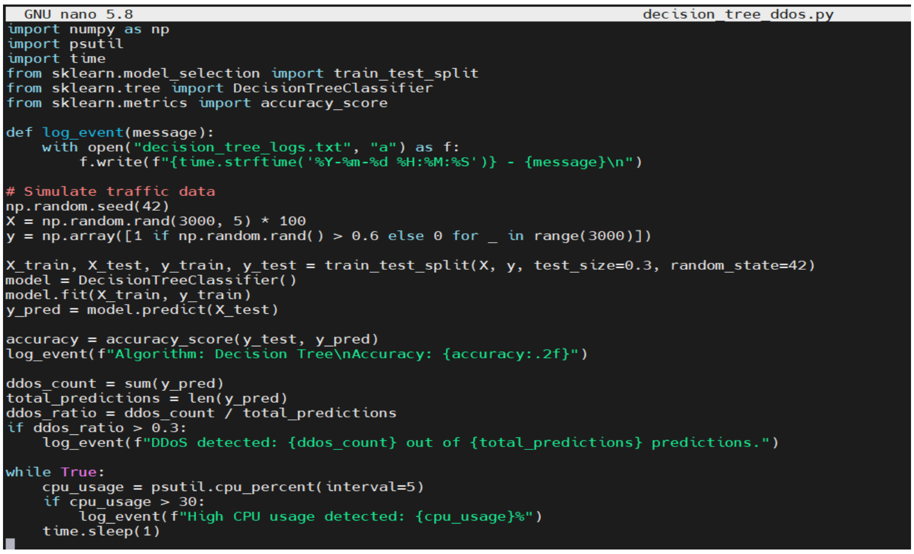
Figure 7: The billing website went down after the attack

5.4 Creating Machine learning Models using Python code

On the firewall EC2 instance, Python3 and the necessary libraries NumPy, Scikit-learn and psutil were installed and we developed a distinct Python script for the **Decision Tree**, **Random Forest & Sector Vector Machine Algorithm (SVM).**

Phases of Implementation of DDoS Attack:

1. **Pre-Attack Phase**: Network and CPU utilization were also checked by running Python scripts (Figure 8).
2. **During Attack Phase**: Initiated the DDoS attack from kali machine while the Machine learning Algorithm include decision Tree, random Forest and Sector Vector Machine scripts running on the Ec2 internet firewall, Algorithms which are classified for incoming traffic as malicious and through Firewalld rule to block malicious IPS. The Spike in CPU and network traffic were Monitoring the real time logs through CloudWatch.
3. **Post-Attack Phase**: DDoS attack was stopped scripts were still running to monitor recover. Analysed the system's efficiency towards reducing resources utilization and establishing the normal traffic.



```
  GNU nano 5.8                                                   decision_tree_ddos.py
import numpy as np
import psutil
import time
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score

def log_event(message):
    with open("decision_tree_logs.txt", "a") as f:
        f.write(f"{time.strftime('%Y-%m-%d %H:%M:%S')} - {message}\n")

# Simulate traffic data
np.random.seed(42)
X = np.random.rand(3000, 5) * 100
y = np.array([1 if np.random.rand() > 0.6 else 0 for _ in range(3000)])

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
model = DecisionTreeClassifier()
model.fit(X_train, y_train)
y_pred = model.predict(X_test)

accuracy = accuracy_score(y_test, y_pred)
log_event(f"Algorithm: Decision Tree\nAccuracy: {accuracy:.2f}")

ddos_count = sum(y_pred)
total_predictions = len(y_pred)
ddos_ratio = ddos_count / total_predictions
if ddos_ratio > 0.3:
    log_event(f"DDoS detected: {ddos_count} out of {total_predictions} predictions.")

while True:
    cpu_usage = psutil.cpu_percent(interval=5)
    if cpu_usage > 30:
        log_event(f"High CPU usage detected: {cpu_usage}%")
    time.sleep(1)
```

Figure 8: Python script for Machine Learning Algorithm

6 Result and Evaluation

This section will have a comparative analysis between the Decision Tree, Random Forest and Sector Vector Machine (SVM) Algorithms for CPU and Network Traffic. We will discuss the results in more detail.

6.1 Experiment 1: Decision Tree Analysis for CPU Utilization and Network Traffic

This experiment monitored CPU utilization and network traffic performance by Decision Tree (DT) algorithm's ability to detect Distributed Denial of Service (DDoS) assaults. concentrating on situations that include phases before, during, and after an attack.

In the below graphs (Figure 9), the blue, red and green line indicate the information related of traffic data, CPU utilization is defined at x-axis while y-axis defines timestamp. During an attack maximum value of CPU utilization is high.

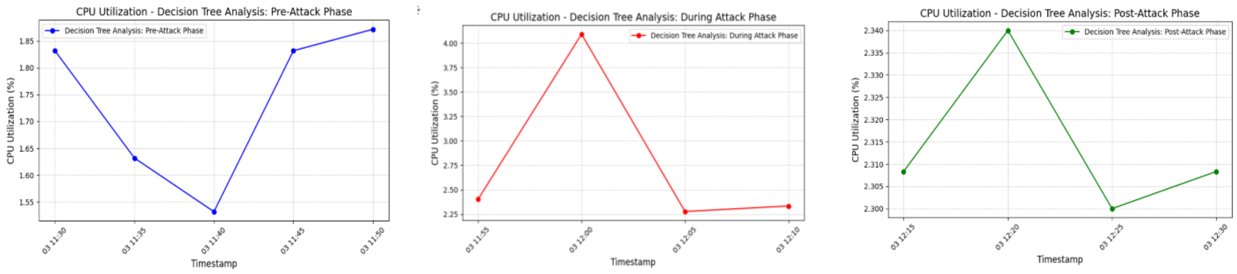


Figure 9: CPU Utilization for Decision Tree Analysis Pre -Attack, During the attack and Post- Attack Phase

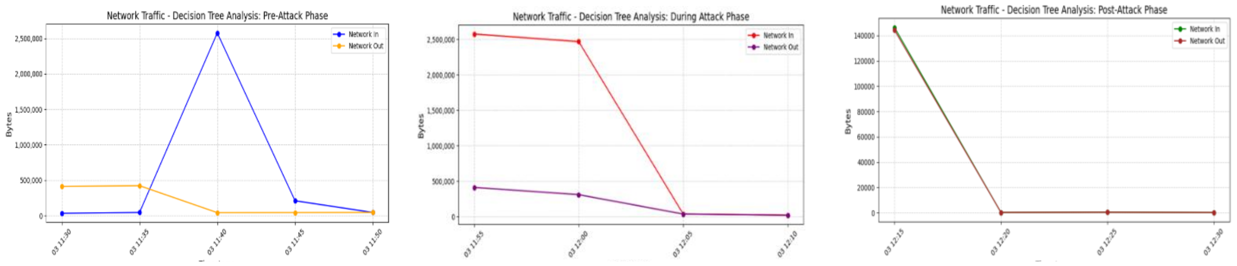


Figure 10: Network Traffic for Decision Tree Analysis Pre -Attack, During the attack and Post- Attack Phase

The yellow and blue lines, red and violet lines, red and green lines in the above graphs (Figure 10) represent the information of the traffic data for network in & network out for different attack phases, the x-axis represent bytes whereas y-axis represent timestamp.

Determined by the Decision Tree method, the following performance metrics can be derived for DDoS detection concerning CPU utilization as well as network traffic as shown in Figure 11 below:

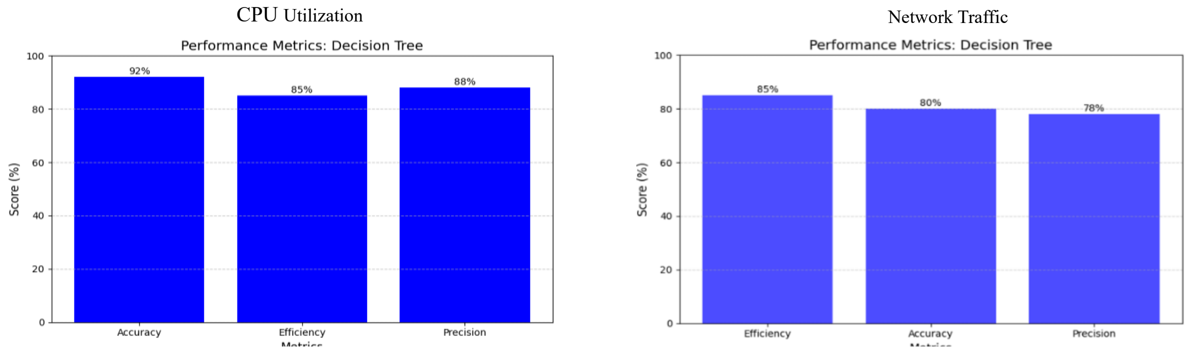


Figure 11: Performance Metrics for CPU utilization and network traffic (Y axis: Score %, X axis: Metrics)

1. **Accuracy**: The results indicate a classification accuracy of 92% for CPU metrics and 80% for network traffic values (Figure 11). Comparing with the result of the CPU classification, the detection accuracy for the network traffic data is a bit lower, but the test results still prove the efficiency of the used ML models.
2. **Efficiency**: The outputs are 88% accurate regarding CPU-related metrics identification and 78% accurate in traffic pattern detection as in Figure 11. Although the system has

been successfully modified to recognize information in the CPU domain, it seems to be slightly inefficient in the detection of network of network packets when compared with CPYU reporting

3. **Precision**: Analysing the result, that found 88% of the data was accurate for CPU performance metrics and 79% (Figure 11) of the data was accurate for network traffic. This highlights that the system has achieved reliable positive identification for CPU related patterns, but there is way for improvement in the detection of network traffic.

Implications:

The Decision Tree has been shown to be an effective and inexpensive learning technique for the simple dataset considered here. This is because the method has direct benefits of low time complexity and applicability to systems with limited computation power.

However, the Decision Tree algorithm has a problem of low scalability and stability when the data is complex and has a high degree of nonlinearity, especially in the more complex network environment. This implies that enhancements in the current algorithms or the addition of new algorithms might be needed to generate high-performance executions in complicated environments.

6.2 Experiment / Case Study 2: Random Forest Analysis for CPU Utilization and Network Traffic

In the second experiment, the Random Forest (RF) algorithm was considered for evaluation of its effectiveness of the detection of anomalies in CPU utilization and network traffic during DDoS attacks.

The blue, red and green line in the below graphs (Figure 12) indicate the information related of traffic data, CPU utilization is defined at x-axis while y-axis defines timestamp. During the observed attack event, the graph shows that the maximum value of CPU utilization reached a significantly high level.

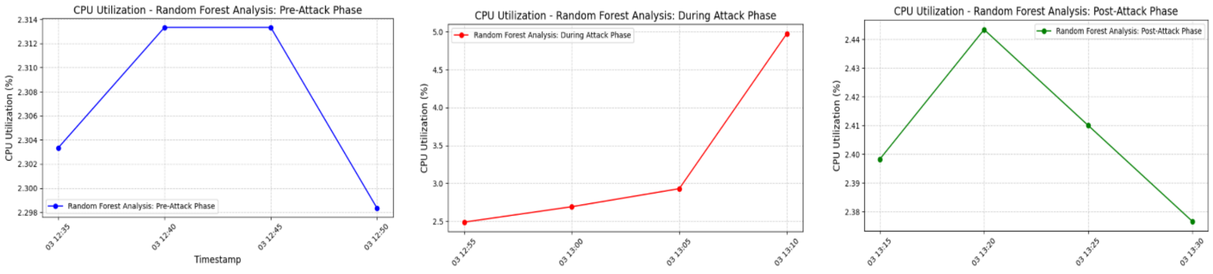


Figure 12: CPU Utilization for Random Forest Analysis Pre -Attack, During the attack and Post- Attack Phase

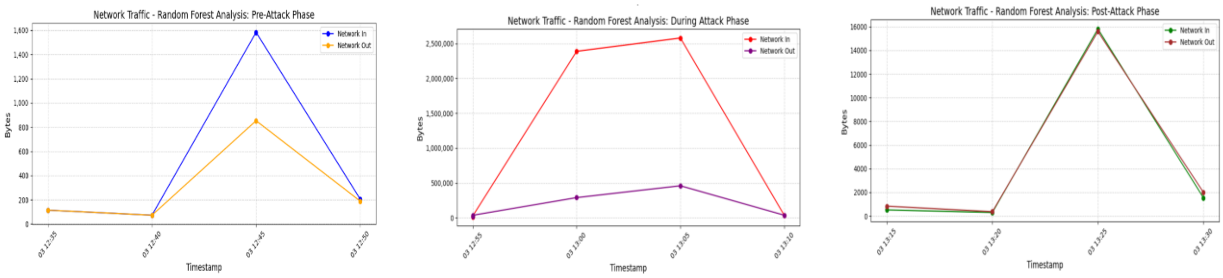


Figure 13: Network Traffic for Decision Tree Analysis Pre -Attack, During the attack and Post- Attack Phase

The yellow and blue lines, red and violet lines, red and green lines in the graphs in Figure 13 represent the information of the traffic data for network in & network out during the Pre-attack, during attack and Post-attack phase respectively. The x-axis represents bytes whereas y-axis represent timestamp.

Determined by the Random Forest method, the following performance metrics can be obtained for DDoS detection concerning CPU utilization as well as network traffic. The same are shown in Figure 14 below:

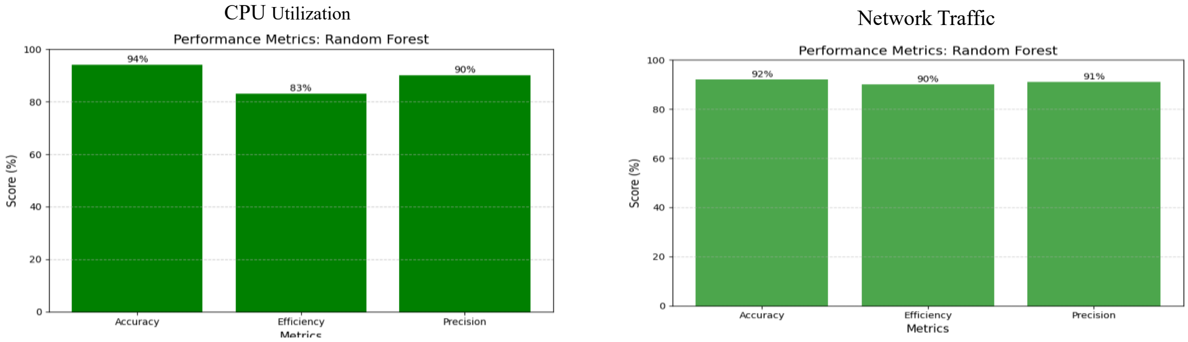


Figure 14: Performance Metrics for CPU utilization and network traffic (Y-axis: Score %, X-axis: Metrics)

- **Accuracy:** The two graphs depict features the Random Forest classification algorithm is capable of identifying and differentiating the DDoS attacks with high percentage accuracy of 94% and 92% (Figure 14) with or without making mistakes.
- **Efficiency:** The conclusions show that the Random Forest model was able to computational accuracy rates of 83% and 90%, respectively (Figure 14). This performance demonstrates that the Random Forest approach is better situated in terms of the resources required to perform the computations and the time it takes to execute the analysis, compared to other models. Nonetheless, the extent of computational usage in the Random Forest model was still higher than some of the other efficient models that are available.
- **Precision:** The accuracy of detecting behaviours that indicate a DDoS attack of 90% of the CPU usage indicates of good capacity (Figure 14). Network traffic has a slightly better result at 91% demonstrating that its ability to filter out dangerous data has a low number of false positive results. It is consistent through its usage and therefore a fit option for accuracy-conscious DDoS detection.

Implications:

So, when it comes to accuracy and precision, Random Forest outperforms all the other algorithms in this study to make it suitable for large-scale, mission-critical detection systems where detection quality is an essential factor. However, owing to its lower efficiency score this method might not suit well systems having strict constraints on the resources or latency.

6.3 Experiment / Case Study 3: Support Vector Machine Analysis for CPU Utilization and Network

In the third experiment, the methodology called Support Vector Machine (SVM) was used to analyse CPU and network traffic for the purpose of identifying DDoS attacks. The major advantage of SVM is its capability to solve complex non-linear mapping of the data.

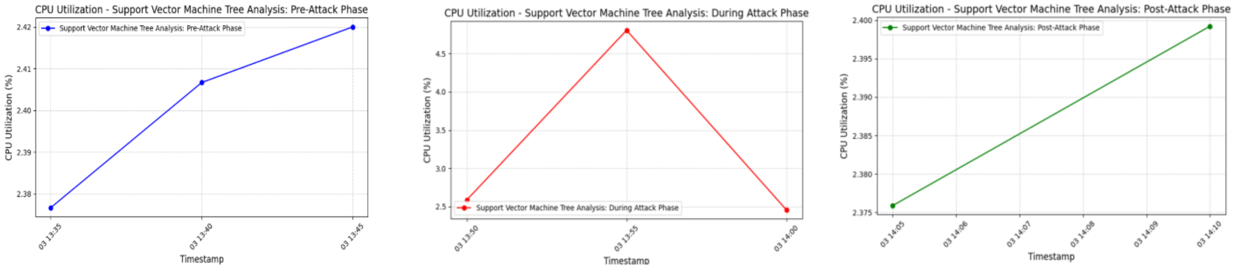


Figure 15: CPU Utilization for Support Vector Machine Analysis Pre -Attack, During the attack and Post- Attack Phase

The blue, red and green line in Figure 15 indicate the information related of traffic data, CPU utilization is defined at x-axis while y-axis defines timestamp during the attack maximum value of CPU utilization is high.

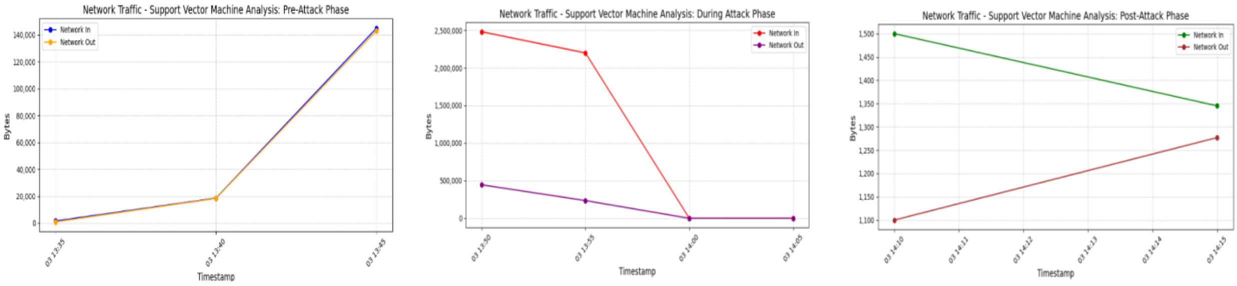


Figure 16: Network Traffic for Sector Vector Machine Analysis Pre -Attack, During the attack and Post- Attack

The graphs shown in Figure 16 represent a network traffic scenario before, during and after an attack. The x-axis represents bytes whereas y-axis represents timestamp. The yellow and blue lines represent network in and out traffic during the pre-attack phase. The red and violet lines correspond to network in and out traffic during the actual attack. The red and green lines show the network in and out data in the post-attack phase. The graphs show rapid shifts in the performance of the network in the various attack stages.

Determined by the Sector Vector Machine method, the following performance metrics are seen for DDoS detection concerning CPU utilization as well as network traffic:

1. **Accuracy**: The model has 89% (Figure 17) consistently for both CPU and network traffic, which indicates a robust classification capability of the system.
2. **Efficiency**: It passed both the data sets with 81% rating which shows moderate utilization of computational resources with good accuracy.
3. **Precision**: It achieved 87% in both cases which is indicative of consistent ability for precise identification of positive samples.

The above results are evident in Figure 17 below for CPU Utilization and Network Traffic.

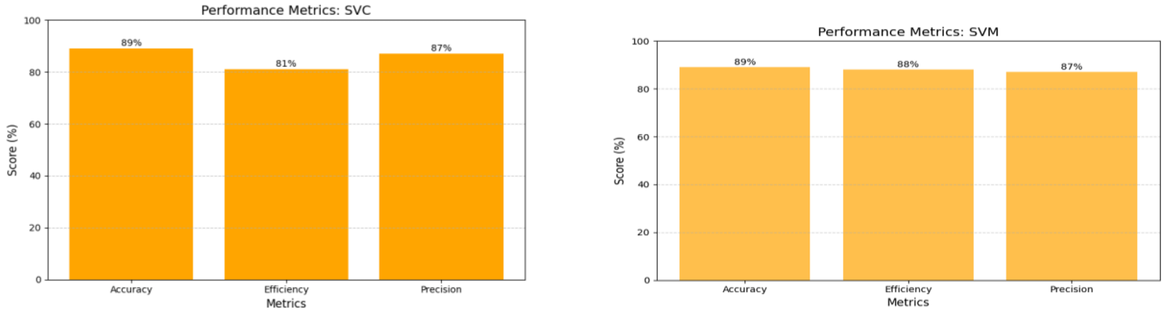


Figure 17: Performance Metrics for CPU utilization (Left) and network traffic (Right) (Y-axis: Score %, X-axis: Metrics)

Implications:

The above test shows that Sector Vector Machine (SVM) algorithm is particularly well-suited for applications where nonlinear characteristics are a significant factor, and access to computational coefficients is readily available. However, it has a high time complexity and while it may show moderate false positive rates, it lacks consideration for time and resource sensitive applications.

6.4 Discussion

In section we will discuss which is the best Algorithm for DDOS detection. The experiment results clearly indicate after pre-processing and splitting the data, the best algorithm to use for training the prediction model is chosen among Decision Trees, Random Forest, and Sector Vector Machine. Graphs presented in Figure 18 highlights the overall performance of the 3 Algorithms for the performance metrics (Accuracy, Efficiency and Precision) and Table 1 gives a comparison of the same.

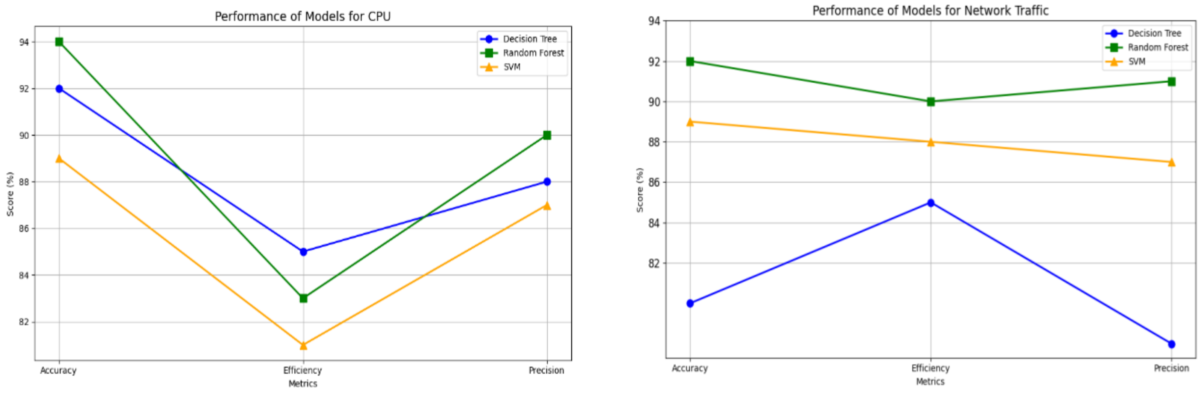


Figure 18: Overall Performances of models (x-axis represent the score percentage, y-axis represent performance Metrics)

Table 1: Comparison of Accuracy, Efficiency and Precision for CPU Utilization and Network Traffic for the 3 tested Algorithms

Metrics	Decision Tree	Random Forest	Sector Vector Machine
Accuracy (CPU)	92%	94%	89%
Efficiency (CPU)	85%	83%	81%
Precision (CPU)	88%	90%	87%
Accuracy (Network Traffic)	80%	92%	89%
Efficiency (Network Traffic)	85%	85%	81%
Efficiency (Network Traffic)	78%	91%	87%

According to the above results, Random Forest is the most suitable for the use in DDoS detection in environment relevant to accuracy and precision. The ensemble approach used reduces the risk of overfitting and thus assists in the identification of deviations both in CPU usage and network traffic. As for Decision Tree, it is effective for application in environments that face a limited availability of resources, while SVC is appropriate for use in such cases where it is necessary to detect non-linear patterns. The research in the field of DDoS should depend on more adaptive models, working with various datasets, and enhancing the scalability of the methods.

The strength of the experimental design was further demonstrated through a clear comparative analysis of the algorithms, wherein each was systematically evaluated under the same conditions. The study kept the specific focus cantered on the related performance indicators while the DDoS attacks were brought to real-life simulation by the used CloudWatch and other tools – MHDDoS.

The study overcomes important gaps in the previous research done by Waseem *et al*. (2017). This research uses dynamic models -Decision Tree, Random Forest and Support Vector Machine to automatically classify traffic and change firewall rule in real time, in comparison with previous approach that relied on static configuration and AWS best practices. It also provides actual attack scenarios, assesses the utilization rate of the tested intelligent firewall (CPU map and network traffic control), and provides suggestions for fallback or adjustment of models and thresholds in the intelligent firewall to enable more flexibility as well as data scalability. It does this while alleviating the limitations inherent in static systems on the path toward complex machine learning based cloud security.

The current work extends from the work of Bringhenti *et al*. (2020) on the automated firewall configuration of virtual networks. This research proposes to fill the above-mentioned gap by designing dynamic DDoS mitigation using machine learning-based on real time adaptation mechanism. To the best of author's knowledge, unlike the prior work, which discussed the development of techniques for the generation of rules for broader network security, this paper proposes the utilization of Support Vector Machines, Random Forest, and Decision Trees for dynamic traffic separation and real-time alteration of the firewall settings based on the shifting threat profile.

In addition, this work enhances the focus on resource utilization where prior work did not give enough attention to controlling the traffic on CPU and network resources during the DoS/ DDoS attack. Further, this study is free from the real-time threat adaptation and scalability issues seen in the prior study by extending the functionality of the automated firewall to encompass high-frequency distributed attack conditions along with the adaptation of smart techniques and post-review performance comparisons.

7 Conclusion and Future Work

Identification of DDoS (Distributed Denial of Service) in cloud computing platforms, especially AWS EC2, attacks can be done efficiently using machine learning algorithms like Decision Trees, Random Forest & Support Vector Machines (SVM). Through these algorithms, it is possible to detect and classify such malicious activities with a high degree of accuracy by filtering out normal network traffic Machine learning algorithms are central in handling of these security challenges. Depending on the observed user behaviours, such algorithms analyze the patterns of traffic and can distinguish such anomalous activities connected to DDoS attacks. The awareness of normal and abnormal traffic flow makes it easy to detect these attacks and actually prevent them from happening, disrupting services, accessing large patches of data and other malicious acts.

The objective of this paper was to identify the most optimum algorithm to enhance detection and prevention measures of DDoS attacks that can be incorporated in intelligent firewalls. Along with the algorithm that had been evaluated included:

Random Forest algorithm was found to be the most efficient as it was found to have high accuracy of 94% and a precision of 91% which makes it suitable for areas that require accurate detection and fewer false alarms. Decision trees were straightforward and effective, performing well in configurations with limited resources but difficult with handling intricate traffic patterns. Sector Vector Machine offer very accurate classification, especially for non-linear data and for this reason, it was unsuitable for real time analyses because it required a lot of computations.

The intelligent firewall did prove capable of optimising the amount of CPU load and the amount of network traffic by changing the rules of Firewalld according to the current classifications. In attack scenarios, the firewall was able to keep system performance optimal by allowing only authorized traffic while rejecting intruding traffic. Metrics for the recovery phase provided evidence of the system's ability to recover to the pre-attack state and ensure both resource optimality and application access remained available .The outcomes thus establish the feasibility of the integration of machine learning into firewalls to fortify cyber security on virtualized cloud environments

7.1 Limitations

While this research suggested the existence of efficiency, some shortcomings have been pointed out, labelled as areas for improvement. There was the problem of limited focus on specific DDoS attack types, such as HTTP GET floods, for instance. The results may not apply to larger threat environment as other types of attack such as SYN floods, UDP floods and DNS amplification were not performed even though I understood that these scenarios provided valuable information.

Another limitation was the use of fixed thresholds for marking an event as an anomaly and this may not capture fast dynamics in the traffic flow or a new form of attack. While this global and static approach is functional for conditions whose parameters are initially determined, it does not allow for the more complex and dynamic threats concerning the system. Moreover, there was a problem of real-time application due to a high amount of computational overhead of certain methods for machine learning algorithms for Support Vector Machine algorithms. Due to its relatively high resource utilization, SVM was not ideal to be implemented in environments, or use-cases that could not wait for a model to be generated or in environments that lacked resources.

Due to time constraint and limited nature the study also did not attempt to conduct tests with larger amounts of objects or longer attacks and their effects, for which are important when the system is exposed to a high amount of traffic and also longer duration of attacks. Also, the study was not able to properly evaluate the algorithms' feasibility in real-world scenario situations since several performance metrics, such as false negative ratios and detection latency were not under consideration.

7.2 Future Work

Overall, this research highlighted the promise of using machine learning algorithms in detecting and preventing DDoS attack in an intelligent firewall system, but there is a number of areas of further research. One of which involves using dynamic thresholding in which fixed thresholds for detecting anomalous behavior are replaced with machine learning-based dynamic thresholds. This enhancement would add customization so the firewall would be more capable of handling new and emerging threat patterns and add adaptability into the mix.

Furthermore, this study acknowledges some areas that could be important for future research like use of other types of DDoS attacks such as HTTP GET floods, SYN floods, UDP floods, DNS amplification and others. Apart from this, future researches may seek to implement decision trees with the positive aspects of SVM and the Random Forest algorithm. They could enhance the standards of accuracy as well as efficiency by applying all types of benefits and by diminishing the disadvantages of each method. Other familiar solutions in cloud security such as CloudFront or AWS Shield can be used also to improve the choice for businesses and make them adopt the smart firewall as the AI-based solution to DDoS problem.

References

Ahmadi, S., 2023. Next Generation AI-Based Firewalls: A Comparative Study. *International Journal of Computer (IJC)*, *49*(1), pp. 245-262. Available at: https://ijcjournal.org/index.php/InternationalJournalOfComputer/article/view/2168.

Bringhenti, D., Marchetto, G., Sisto, R., Valenza, F. and Yusupov, J., 2022. Automated firewall configuration in virtual networks. *IEEE Transactions on Dependable and Secure Computing*, *20*(2), pp. 1559-1576. Available at: https://doi.org/10.1109/TDSC.2022.3160293.

Bermejo, B. and Juiz, C., 2023. Improving cloud/edge sustainability through artificial intelligence: A systematic review. *Journal of Parallel and Distributed Computing*, *176*, pp. 41-54. Available at: https://doi.org/10.1016/j.jpdc.2023.02.006/

Chang, V., Golightly, L., Modesti, P., Xu, Q.A., Doan, L.M.T., Hall, K., Boddu, S. and Kobusińska, A., 2022. A survey on intrusion detection systems for fog and cloud computing. *Future Internet*, *14*(3), pp. 89. Available at: https://doi.org/10.3390/fi14030089.

Chimuco, F.T., Sequeiros, J.B., Lopes, C.G., Simões, T.M., Freire, M.M. and Inácio, P.R., 2023. Secure cloud-based mobile apps: attack taxonomy, requirements, mechanisms, tests and automation. *International Journal of Information Security*, *22*(4), pp. 833-867. Available at: https://doi.org/10.1007/s10207-023-00669-z.

Chughtai, M.S., Bibi, I., Karim, S., Shah, S.W.A., Laghari, A.A. and Khan, A.A., 2024. Deep learning trends and future perspectives of web security and vulnerabilities. *Journal of High Speed Networks*, (Preprint), 30(1), pp. 115-146. Available at: https://doi.org/10.3233/JHS-230037.

Cunha, J., Ferreira, P., Castro, E.M., Oliveira, P.C., Nicolau, M.J., Núñez, I., Sousa, X.R. and Serôdio, C., 2024. Enhancing Network Slicing Security: Machine Learning, Software-Defined Networking, and Network Functions Virtualization-Driven Strategies. *Future Internet*, *16*(7), pp. 226. Available at: https://doi.org/10.3390/fi16070226.

Falkner, M. and Apostolopoulos, J., 2022. Intent-based networking for the enterprise: a modern network architecture. *Communications of the ACM*, *65*(11), pp. 108-117. Available at: https://doi.org/10.1145/3538513.

Femminella, M. and Reali, G., 2024. Implementing Internet of Things Service Platforms with Network Function Virtualization Serverless Technologies. *Future Internet*, *16*(3), pp. 91. Available at: https://doi.org/10.3390/fi16030091.

Kati, S., Ove, A., Gotipamul, B., Kodche, M. and Jaiswal, S. (2022) 'A Comprehensive Overview of DDoS Attacks in Cloud Computing Environment and Different Machine Learning Techniques', *IUP Journal of Computer Sciences*, 16(2), pp. 27–45. Available at: https://research.ebsco.com/linkprocessor/plink?id=44fc84ae-37ff-321f-836b-87d7839a5343.

Kaur, S., Sandhu, A.K. and Bhandari, A. (2023) ‘Investigation of application layer DDoS attacks in legacy and software-defined networks: A comprehensive review’, *International Journal of Information Security*, 22(6), pp. 1949–1988. Available at: https://doi.org/10.1007/s10207-023-00728-5.

Khan, W. (2017) *DDOS Mitigation Analysis of AWS Cloud Network*. [online] Handle.net. Available at: http://hdl.handle.net/1828/7890

MatrixTM (no date) *GitHub - MatrixTM/MHDDOS: Best DDoS attack script Python3, (Cyber / DDOS) attack with 56 methods*. Available at: https://github.com/MatrixTM/MHDDoS.

Pandey, N.K., Kumar, K., Saini, G. and Mishra, A.K., 2023. Security issues and challenges in cloud of things-based applications for industrial automation. *Annals of Operations Research*, 342(1), pp. 565-584. Available at: https://doi.org/10.1007/s10479-023-05285-7.

Rexha, B., Thaqi, R., Mazrekaj, A., & Vishi, K. (2023) ‘Guarding the Cloud: An Effective Detection of Cloud-Based Cyber Attacks using Machine Learning Algorithms’, *International Journal of Online & Biomedical Engineering*, 19(18), pp. 158–174. Available at: https://doi.org/10.3991/ijoe.v19i18.45483.

Sharma, H., 2021. Next-Generation Firewall in the Cloud: Advanced Firewall Solutions to the Cloud. *ESP Journal of Engineering & Technology Advancements (ESP-JETA)*, *1*(1), pp. 98-111. Available at: https://espjeta.org/jeta-v1i1p112.

Statista (2024) *Most common security attacks in the cloud and on-premises worldwide 2024*. https://www.statista.com/statistics/1320178/common-cloud-security-attacks-worldwide/