Online-Journals.org

iJOE International Journal of Online and Biomedical Engineering

iJOE | eISSN: 2626-8493 | Vol. 19 No. 18 (2023) | OPEN ACCESS

https://doi.org/10.3991/ijoe.v19i18.45483

PAPER

Guarding the Cloud: An Effective Detection of Cloud-Based Cyber Attacks using Machine Learning Algorithms

Blerim Rexha[1], Rrezearta Thaqi[1](✉), Artan Mazrekaj[1], Kamer Vishi[2]

[1]Faculty of Electrical and Computer Engineering, University of Prishtina, Prishtina, Kosovo

[2]Department of Informatics, University of Oslo, Oslo, Norway

rrezearta.thaqi@uni-pr.edu

ABSTRACT

Cloud computing has gained significant popularity due to its reliability and scalability, making it a compelling area of research. However, this technology is not without its challenges, including network connectivity dependencies, downtime, vendor lock-in, limited control, and most importantly, its vulnerability to attacks. Therefore, guarding the cloud is the objective of this paper, which focuses, in a novel approach, on two prevalent cloud attacks: Distributed Denial-of-service (DDoS) attacks and Man-in-the-Cloud (MitC) computing attacks. To tackle the detection of these malicious activities, machine learning algorithms, namely Decision Trees, Support Vector Machine (SVM), Naive Bayes, and K-Nearest Neighbors (KNN), are utilized. Experimental simulations of DDoS and MitC attacks are conducted within a cloud environment, and the resultant data is compiled into a dataset for training and evaluating the machine learning algorithms. The study reveals the effectiveness of these algorithms in accurately identifying and classifying malicious activities, effectively distinguishing them from legitimate network traffic. The finding highlights Decision Trees algorithm with most promising potential of guarding the cloud and mitigating the impact of various cyber threats.

KEYWORDS

cloud computing, decision trees, support vector machine, naive bayes, k-nearest neighbors, machine learning, attacks, security

1 INTRODUCTION

Over the past years, cloud computing has been one of the most popular and fast growing technologies. It provides a range of different services for various applications such as data storage, servers, databases, networking, and software [1]. As it is combined by Internet, distributed systems and virtualization, it allows the users

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 and Biomedical Engineering (iJOE)*, 19(18), pp. 158–174. https://doi.org/10.3991/ijoe.v19i18.45483

Article submitted 2023-09-07. Revision uploaded 2023-10-24. Final acceptance 2023-10-26.

to access the information technology infrastructure and applications on demand through the Internet. Compared to locally deployed information technology applications and solutions, cloud computing is characterized by virtualization, dynamic and high scalability, on-demand deployment, and high flexibility.

Cloud computing has changed the understanding and functionality of the applications. This includes stronger computing power at a lower cost and also the combination with artificial intelligence, Internet of Things, and machine learning to enhance the scope of applications.

The growth in cloud-enabled services and cloud market is unprecedented in the today's world. According to [2], this growth was made rapidly year by year, as clearly presented in Figure 1. In 2020, that growth was only 6.1%, but the section of the market that's seen the most growth is Desktop as a Service (DaaS), which grew by 95.4% in 2020, 61.5% in 2021 and 30.7% in 2022. Detailed statistics about cloud-enabled services and cloud market are presented in Figure 1.

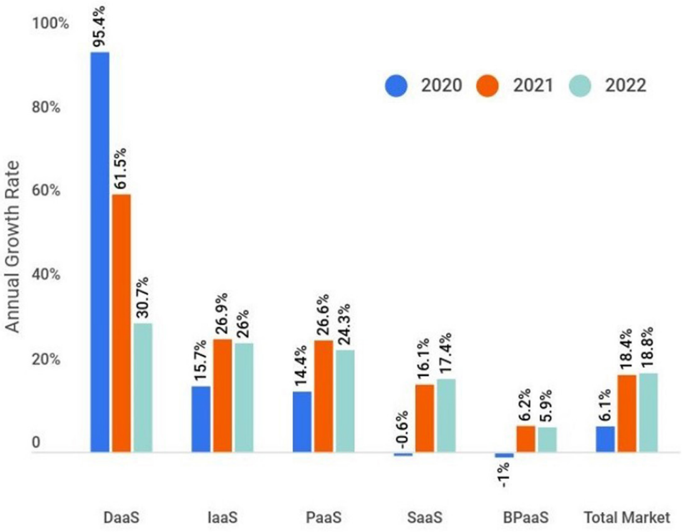


Fig. 1. Public cloud services annual growth rate [2]

As it is expanding very fast together with improved efficiency, improved scalability and faster deployments, cloud computing has many things to worry about such as data security, privacy, and the development of cloud services. Despite the different advantages that it has, cloud computing has also several disadvantages, and one of those is the vulnerability to attacks.

In recent years there were many attacks towards cloud environments, especially Distributed Denial of Service (DDoS) and authentication attacks. DDoS attacks are one of the most common cybercrime attacks and an important challenge in cloud computing because of the loss of trust and privacy that can lead. As in cloud, security is an important topic, the prevention of this kind of attack is also very important as this attack can generate a lot of extra network traffic that makes it difficult for the server to identify potential threats and traffic [3]. On the other hand, authentication attacks are very popular. According to [4], from 2020 to 2022 after phishing, different malware attacks, data loss, etc., have grown day by day. As presented in Figure 2, there is an increasing trend of attacks in cloud.

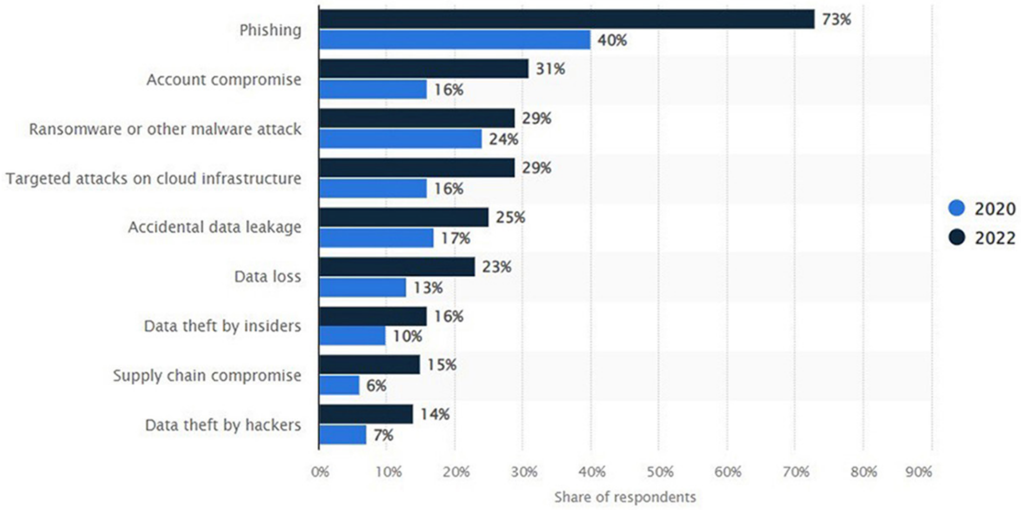


Fig. 2. Most common cloud security incidents worldwide in 2020 and 2022 [4]

Today, many researchers show their interest in analyzing these attacks through different methods. One of these methods that is of great interest to the researcher is machine learning. Machine learning is one of the most popular fields that is being used for attacks detection due to the precise results it gives. In the machine learning-based security system, abnormal and healthy behaviors are categorized based on different labels using the training models. Algorithms of machine learning were applied to biometric authentication, as presented by [5] and [6], or to enhance the feature list of standard security tools, such as Burp Suite, as explained by [7]. Machine learning models ensure the security of the cloud environment by extracting features of different types that help to detect attacks. Although the machine learning techniques confront the different attack types over the abundant cloud environment, it fails to recognize unknown attacks [8].

The main contribution of this paper is to detect the most common attacks that happen in cloud: Distributed Denial-of-service (DDoS) attack and Man-in-the-Cloud (MitC) computing attack, with the help of machine learning algorithms.

The rest of the paper is structured as follows: Section 2 presents the related work in machine learning for cloud attacks detection; Section 3 presents the security vulnerabilities in cloud computing (DDoS and MitC) describing each of them; Section 4 presents the implementation process and results, and Section 5 wraps up the paper with a conclusion and indication of future work.

2 RELATED WORK

2.1 Machine learning approaches for cloud attacks detection

Cloud attacks detection has been a popular topic for a very long time, and it still is. There are many research papers that proposed different solutions to solve this problem. Different papers approached this problem by using different methods such as deep learning, machine learning, artificial intelligence, and many other methods. The methods mentioned seemed to give good results with high precision.

Nathezhtha and Yaidehi [9] proposed an Improvised Long Short-Term Memory (ILSTM) model that trains itself automatically and learns the behavior of the user.

The model proposed classifies the user behavior in two categories: normal or abnormal. This model not only does the identification of the anomaly nodes but also finds whether a misbehaving node is a broken node. Except that the model detects the attack with a high accuracy it also reduces the false alarm in the cloud network.

Sharma et al. [10] analyzed and proposed the machine learning algorithms for detecting DDoS attack in cloud computing environment. They used the technique of isolation forest anomaly detection and then, to detect the DDoS attack, the correlation method was used.

Manna and Alkasassbeh [11] presented an approach that used machine learning algorithms, such as decision tree J48, random forest, and REP tree. The proposed technique used SNMP-MIB data for the trained Intrusion Detection System (IDS) to detect DOS attack anomalies that may affect the network. The results showed that applying the Reduced Error Pruning (REP) tree algorithm classifier donated the highest performance to all IP set times. The average performance of these three classifiers was accurate enough to be an IDS System.

Singh [12] presented a method which combines statistical and machine learning methods to efficiently detect and mitigate DDoS attacks in SDN – Software Defined Network. The method presented seemed to achieve a very high accuracy (of 99.26%) and a detection rate of 100% in detecting and mitigating DDoS attacks in a software defined network.

An interesting approach was conducted by [13]. In this research there was a hybrid algorithm designed that consists of a combination of several machine learning techniques to train a model that can be used to detect and to classify the type of DDoS attack with an accuracy that is greater than the accuracy of each individual machine learning technique used in the hybrid model.

Mishra et al. [1] proposed the perplexed Bayes classifier for DDoS detection in the cloud. The authors used performance parameters, perplexed-based classifiers with and without feature selection and compared them with Naive Bayes and Random Forest to compare the accuracy between them. They proved that the algorithm they proposed has an accuracy of 99% and that the algorithm is very efficient in detecting DDoS attacks in different systems that are in cloud.

Arif and Nassif [3] introduced a system to detect and prevent DDOS attacks based on the analysis of the characteristics of incoming packets to the network, and train and classify the system through machine learning algorithms based on several extracted features that clearly affect the process of data flow. The algorithm used was the SVM algorithm that was comparable to neural networks in accuracy and efficiency. The results showed that the SVM can detect the attack with 99% efficiency and in less time.

Zekri et al. [14] proposed a system for detecting DDoS attacks based on the algorithm C4.5 to mitigate the DDoS threat. The proposed algorithm generates a decision tree to detect signature attacks automatically and effectively for DDoS flooding attacks when used in conjunction with signature detection techniques. The system focuses more on flood-based attack targeting layers 3 and 4 in the 7-layer OSI model. The results show that the proposed approach through the adaptation of the c4.5 algorithm produces results with higher accuracy than other machine learning techniques.

The authors in [15] have proposed a detection framework that uses an ML model to power IDS to detect network traffic anomalies. The dataset used by the detection model includes both malicious and normal traffic and contains two types of features, the first is extracted from the network traffic flow and the other is computed for a specific period. Six machine learning techniques are used to train the model: ANN, KNN, DTREE, SVM, Naive Bayes and Random Forest, which are then tested using the cross-validation and split-validation methods.

3 SECURITY VULNERABILITIES IN CLOUD COMPUTING

Nowadays, hackers take advantage of the cloud computing service to conduct illegal activities in a distributed environment. With the increased computing capability of cloud services, hackers launch attacks in a short period. In this section we will show some of the most common attacks in cloud computing [8]. In the cloud environment, security threats occur from within the organization as well as outside the organization [16]. Possible vulnerabilities can be created by a malicious insider in the cloud environment.

3.1 DDoS attacks

A Distributed Denial-of-Service (DDoS) attack is a malicious attempt to disrupt the normal traffic of a targeted server, service or network by overwhelming the target or its surrounding infrastructure with a flood of Internet traffic. DDoS attacks achieve effectiveness by utilizing multiple compromised computer systems as sources of attack traffic [17]. Exploited machines can include computers and other networked resources such as IoT devices. From a high level, a DDoS attack is like an unexpected traffic jam clogging up the highway, preventing regular traffic from arriving at its destination.

A DDoS attack is commonly characterized as an event in which a legitimate user or organization is deprived of certain services, like web, email or network connectivity, that they would normally expect to have. DDoS is basically a resource overloading problem. The resources can be bandwidth, memory, CPU cycles, file descriptors, buffers etc. The attackers bombard scarce resources either by flood of packets or a single logic packet which can activate a series of processes to exhaust the limited resource. Figure 3 illustrates a simplified DDoS attack scenario, where the attacker uses three zombies to generate a high volume of malicious traffic to flood the victim over the Internet, thus rendering legitimate users unable to access the service [18].

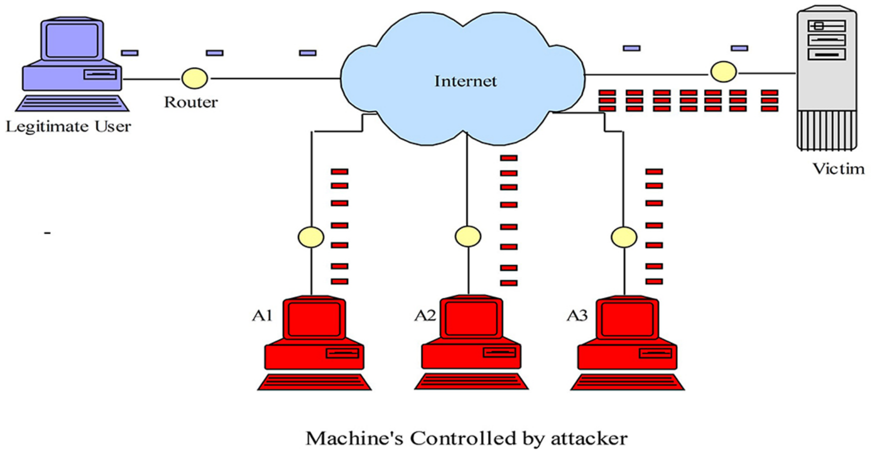


Fig. 3. DDoS attack scenario [18]

3.2 Man in the cloud attack

One malicious tactic that has become quite prevalent in recent years is known as a Man in the Cloud (MitC) attack. This attack aims to access victims' accounts without the need to obtain compromised user credentials beforehand [19]. To gain access to cloud accounts, MitC attacks take advantage of the OAuth synchronization token system used by cloud applications. Most popular cloud services – Dropbox, Microsoft OneDrive, Google Drive, and more – each save one of these tokens on a user's device after initial authentication is completed. This is done to improve usability – users don't have to enter their password every time they attempt to access an app if they have an OAuth token [19].

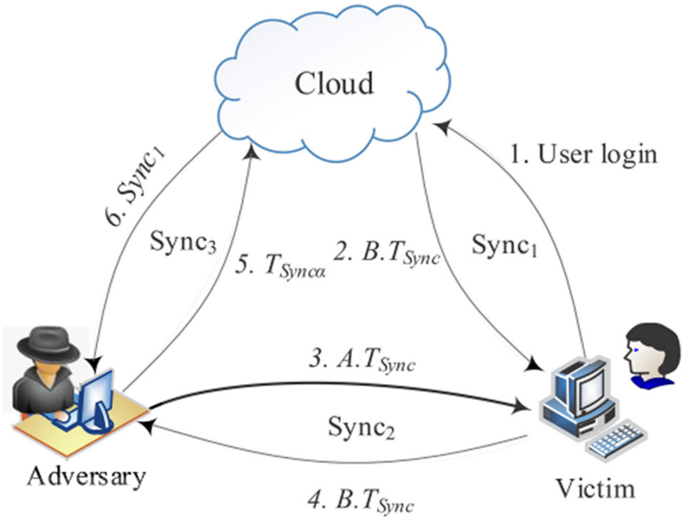


Fig. 4. MitC attack flow [20]

However, anytime, anywhere nature of cloud services means that the same token can grant access from any device. As such, if an attacker can access and copy a token, she or he can infiltrate the victim's cloud remotely – in a manner that appears genuine and bypasses security measures. Zimba and Wang [20] describe the process flow of the MitC attack, as illustrated in Figure 4.

Once executed on the victim's device, this malware installs a new token (belonging to a new account that the attacker created) and moves the victim's real token into a cloud sync folder. Then, when the victim's device next syncs, it syncs the victim's data to the attacker's account instead of the victim's [19].

4 DETECTION OF DDOS AND MITC ATTACKS IN THE CLOUD ENVIRONMENT

4.1 Problem description

Cloud is becoming the go-to solution for companies looking to cut costs and improve efficiency by moving their infrastructure and services to the cloud. However, with the increased use of cloud computing-based services comes a risk which is the increase in cyber-attacks, especially DDoS and MitC attacks. These attacks can significantly impact the availability and security of cloud-based services, resulting in financial loss, reputational damage, and potential legal implications.

One of the main problems that will be addressed in this topic is simulating DDoS and MitC attacks against a cloud environment, then storing the results in a dataset that will be used by machine learning algorithm models in order to make predictions about attacks. Based on the analysis of the results, the cases where the traffic caused to the cloud environment is legitimate or not will be examined. What will be considered for analysis are the number of packets per second, the size of the sent packet and the sequence number.

Traditional methods used to detect attacks may not be as effective nowadays for dynamic cloud environments, which require more sophisticated techniques such as the use of machine learning algorithms. Therefore, there is a need to explore the use of machine learning algorithms for detecting DDoS and MitC attacks in cloud environments.

This topic aims to analyze the effectiveness of machine learning algorithms, including: (i) Decision Trees, (ii) Support Vector Machine (SVM), (iii) Naive Bayes, and (iv) K-Nearest Neighbors (KNN), for detecting DDoS and MitC attacks in cloud environments. The topic will evaluate the performance of each algorithm using a dataset generated by simulating attacks, focusing on accuracy. With the help of the best chosen algorithm, the prediction will be made when these attacks are most likely to occur in a cloud environment.

Moreover, this paper explores the impact of different features, such as the number of requests, packet size and sequence number on the performance of the algorithms and addresses the challenges and limitations of the proposed methods, such as the need for a large amount of training data.

The results are expected to demonstrate the potential of machine learning algorithms for detecting DDoS and MitC attacks in cloud environments. The findings will be useful to cloud service providers and security professionals who are working to improve the security and availability of cloud-based services. Ultimately, this topic will contribute to improving the understanding of machine learning techniques for detecting cyber attacks in cloud environments, thus helping to mitigate and detect risks associated with cloud services.

The source code for the developed application can be accessed via GitHub (https://github.com/Rrezeartaa/DDoSandMitCDetector).

4.2 Experimental evaluation

Cloud environment setup. Amazon Elastic Compute Cloud (EC2) is a part of Amazon's cloud computing platform, Amazon Web Services (AWS), that enables users to create and run their own computer applications on virtual computers. EC2 encourages scalable deployment of applications by providing a web service through which a user can use an Amazon Machine Image (AMI) to configure a virtual machine, which Amazon calls an "instance", containing any desired software. A user can create, launch, and shut down instances that are like servers as needed [21].

Due to its versatile capabilities, Amazon EC2 was used to establish the cloud-based testing environment. The EC2 instance was configured to be externally reachable, hosting a straightforward application. Figure 5 depicts the architectural layout of the generated EC2 instance.

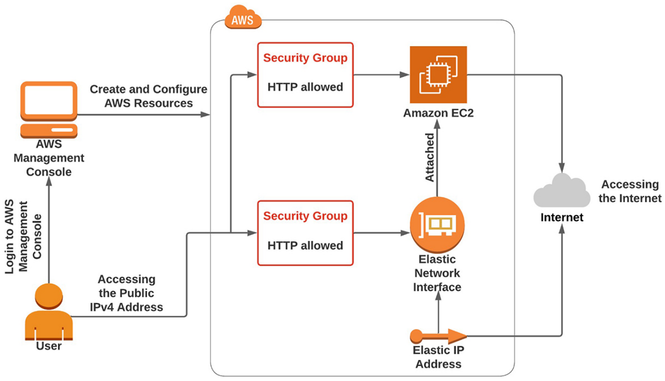
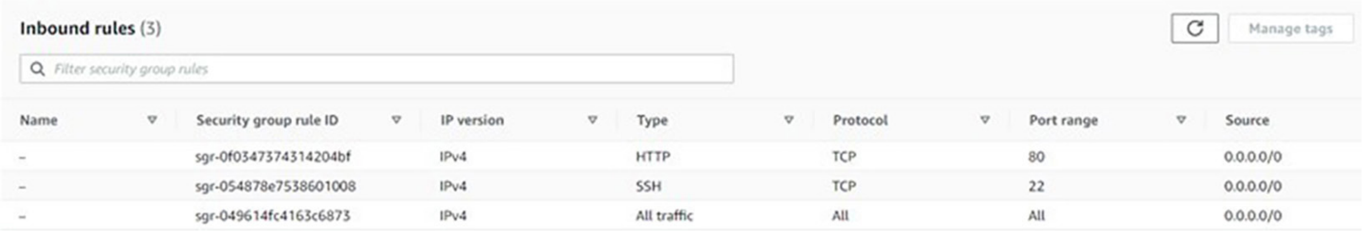


Fig. 5. The EC2 instance architecture [22]

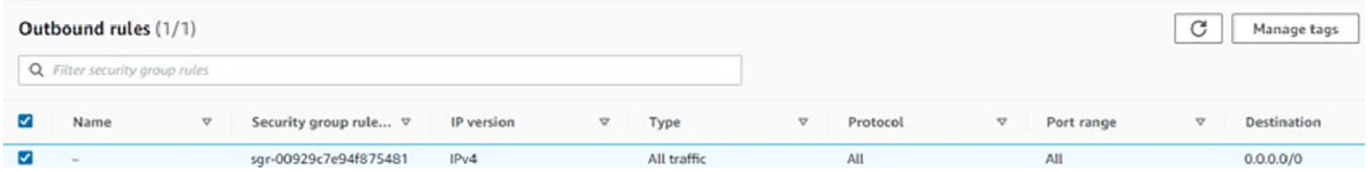
During the experimental setup we have set the security group rules as they're seen in Figures 6 and 7, configuring what can be let in and out of the instance where the application is hosted.



Inbound rules (3)

Name	Security group rule ID	IP version	Type	Protocol	Port range	Source
–	sgr-0f0347374314204bf	IPv4	HTTP	TCP	80	0.0.0.0/0
–	sgr-054878e7538601008	IPv4	SSH	TCP	22	0.0.0.0/0
–	sgr-049614fc4163c6873	IPv4	All traffic	All	All	0.0.0.0/0

Fig. 6. Inbound rules of the instance security group



Outbound rules (1/1)

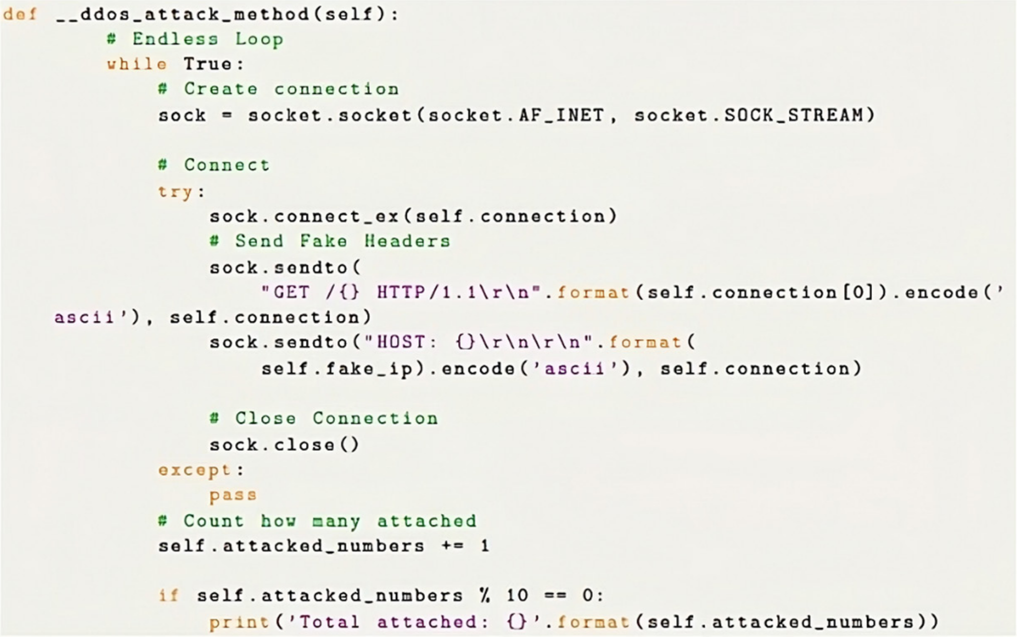
☑	Name	Security group rule...	IP version	Type	Protocol	Port range	Destination
☑	–	sgr-00929c7e94f875481	IPv4	All traffic	All	All	0.0.0.0/0

Fig. 7. Outbound rules of the instance security group

DDoS and MitC attacks simulations. The simulations of DDoS and MitC attacks are done for testing purposes only for this research.

The DDoS attack aims to interrupt or disrupt the availability of the target, which can be an IP, website or even a service, by overloading it with a large volume of traffic generated by a large number of sources, while the MitC attack enables gaining unauthorized access to data. In this implementation for the DDoS attack, the script creates many connections to the target using the specified number of threads that are given in the input and sends a large number of HTTP requests to simulate or create a busy traffic, thus making the cloud environment unavailable in our case, while for the MitC attack, it tries to gain unauthorized access to the instance files.

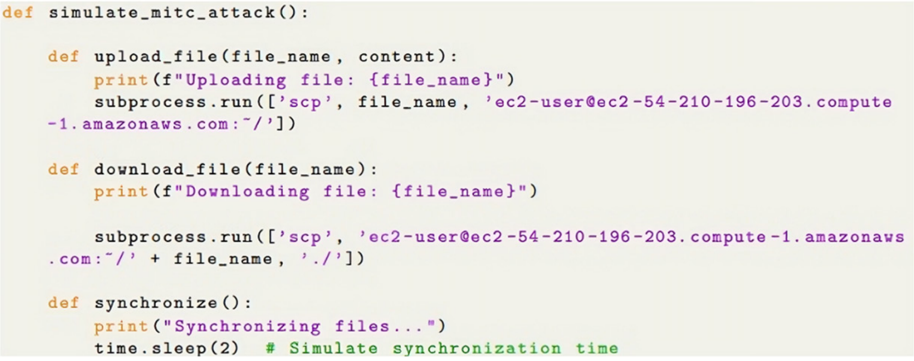
The `__ddos_attack_method()` method is the core part of the DDoS attack, which is executed by each thread in an infinite loop. In this method, it first creates a socket and a TCP connection for the target, along with the fake IP address, in order to simulate requests. Then fake HTTP GET requests are sent to avoid detection. Once the bogus requests/headers are sent, the connection is closed, and the process repeats over and over in an infinite loop, as presented in Figure 8.



```
def __ddos_attack_method(self):
        # Endless Loop
        while True:
            # Create connection
            sock = socket.socket(socket.AF_INET, socket.SOCK_STREAM)

            # Connect
            try:
                sock.connect_ex(self.connection)
                # Send Fake Headers
                sock.sendto(
                    "GET /{} HTTP/1.1\r\n".format(self.connection[0]).encode('
    ascii'), self.connection)
                sock.sendto("HOST: {}\r\n\r\n".format(
                    self.fake_ip).encode('ascii'), self.connection)

                # Close Connection
                sock.close()
            except:
                pass
            # Count how many attached
            self.attacked_numbers += 1

            if self.attacked_numbers % 10 == 0:
                print('Total attached: {}'.format(self.attacked_numbers))
```

Fig. 8. The method for DDoS attack simulation

For the MitC attack, the code has been implemented in a way that unauthorized actions can be performed on the instance, such as accessing and modifying instance files. To do this, the code has been implemented to authorize the instance after needing credentials. Then it is attempted to upload a file to the instance and then download it so that it can be modified and uploaded again with the changes, as presented in Figure 9.



```
def simulate_mitc_attack():

    def upload_file(file_name, content):
        print(f"Uploading file: {file_name}")
        subprocess.run(['scp', file_name, 'ec2-user@ec2-54-210-196-203.compute
    -1.amazonaws.com:~/'])

    def download_file(file_name):
        print(f"Downloading file: {file_name}")

        subprocess.run(['scp', 'ec2-user@ec2-54-210-196-203.compute-1.amazonaws
    .com:~/' + file_name, './'])

    def synchronize():
        print("Synchronizing files...")
        time.sleep(2)  # Simulate synchronization time
```

Fig. 9. A part of code for MitC attack simulation

Packet capturing. The application is implemented in Flask[1]. Its main part has packet capturing, which is the first step for the attacks detection.

The main part in the application is the detection of DDoS and MitC attacks by analyzing the network traffic. This happens when the IP address is given in the application form. The given IP address is used as a filter to determine which packets to process.

The first step is to get the value of the IP address given in the form. This is then sent at the backend side.

Then a list of fields extracted from each packet is defined. These fields include: *duration, source_ip, destination_ip, protocol, syn_bit, ack_bit, fin_bit, packet_size, pps,*

[1] https://www.fullstackpython.com/flask.html

sequence_number and is_attack. These fields present important information about network traffic that helps detect DDoS and MitC attacks.

The *duration* field denotes the time the packet was captured; *source_ip* and *destination_ip* fields reveal sender and receiver addresses, while *protocol* field specifies the communication rules. The *syn_bit*, *ack_bit*, and *fin_bit* fields are control bits to manage connections. The *packet_size* field presents data volume, and *pps* field measures transmission rate (number of packets per second). The *sequence_number* field aids in ordered data exchange. The *is_attack* flag indicates malicious intent that is determined based on the value of the *pps* field (if the value of *pps* is >1000 it is labeled as 1). After defining the fields, the method opens a file called *ddos_mitc_traffic_app.csv* in write mode. This file will be used to store the packet analysis results.

Then, each packet is handled. First, it is checked if the packet is of the specified IP address by looking for the IP protocol in the packet and comparing the source and destination IPs to the IP address used for filtering. If the packet matches the IP address for filtering, the method continues with extracting the desired fields from the packet.

```
start_time = time.time()
packet_size = len(packet)

time.sleep(random.uniform(0, 0.1))

sequence_number = packet[IP].seq

end_time = time.time()
duration = end_time - start_time
pps = packet_size / duration
is_attack = 0

if pps > 1000:
    if previous_seq_number is not None and sequence_number !=
    previous_seq_number + 1:
       is_attack = 1

previous_seq_number = sequence_number
```

Fig. 10. Extracting the desired fields from the packet

Next, before processing the data from the packet, a start time is set (the time when the packet was received) and at the end the time ,when the packet was processed. By subtracting these times, it is found how long the processing period of the package has been. The start time of each packet is recorded and the packet is delayed for a random time between 0 and 0.1 seconds using the `time.sleep()` function so that the simulated attack looks as realistic as possible. The packet size and the number of packets per second (*pps*) are then calculated. If the value of *pps* exceeds 1000, it checks the sequence number for that packet and the packet is considered illegitimate if it meets the condition given in the code and the value of the *is_attack* variable becomes 1; otherwise, it remains 0, as presented in Figure 10.

4.3 Dataset integration and model training

Dataset integration. To integrate the results obtained from the packet analysis, the following steps have been taken:

1. **Collection of results and data storage:** after running the packet analysis script, they are stored in CSV file as shown in the section above.
2. **Dataset cleaning and formatting:** Python's pandas library is used to clean and format data such as removing unnecessary columns and converting columns to the correct data type.

3. **Splitting the data set:** the train test split function from the scikit-learn library is used to split the data into a training set and a test set.
4. **Finding the best algorithm:** the scikit-learn library is used to select a machine learning model that is suitable for the classification task. In our case, the metrics of four machine learning algorithms: (i) Decision Trees, (ii) SVM, (iii) Naive Bayes and (iv) KNN were compared, and based on accuracy, the best algorithm for training the model is selected.
5. **Training and evaluation of the model:** using the adaptation method ('fit' method). the selected model in the training set is trained.
6. **Model evaluation:** the prediction method (the 'predict' method) is used to make predictions in the test group and evaluation metrics such as accuracy and precision are used to evaluate the performance of the model.
7. **Model deployment:** once the performance of the model has been evaluated, it can be deployed to classify the results of a DDoS and MitC attack.

Finding the best algorithm and training the best model. After pre-processing and splitting the data, the best algorithm to use for training the prediction model is chosen among Decision Trees, Naive Bayes, SVM and KNN. These four algorithms were taken into consideration because of the distinct characteristics and versatility they have.

The goal is to find the best algorithm immediately after analyzing the packets to make predictions in seconds or even milliseconds.

First, the models are defined and added to a "models" list. Then two empty lists "results" and "names" are created in which the accuracy values for each model are stored. To obtain the accuracy values for each model of the algorithm, a for loop has been created which goes through all the algorithm models stored in the "models" list. A portion of the CSV file used for training the models is presented in Table 1:

Table 1. A section of the CSV file extracted from the code's output

Duration	Source_ip	Destination_ip	Syn_bit	Ack_bit	Fin_bit	Packet_size	PPS	Sequence_number	Is_attack
0.06654	54.210.196.203	192.168.1.17	FALSE	TRUE	FALSE	60	901.693	0	0
0.10388	54.210.196.203	192.168.1.17	FALSE	TRUE	FALSE	60	577.542	0	0
0.10402	54.210.196.203	192.168.1.17	FALSE	TRUE	FALSE	60	576.771	0	0
0.04623	54.210.196.203	192.168.1.17	FALSE	TRUE	FALSE	60	1297.789	0	1
0.01427	54.210.196.203	192.168.1.17	FALSE	TRUE	FALSE	60	4202.639	0	1
0.05934	54.210.196.203	192.168.1.17	FALSE	TRUE	FALSE	60	1011.093	0	1
0.04503	54.210.196.203	192.168.1.17	FALSE	TRUE	FALSE	60	1332.208	0	1
0.04585	192.168.1.17	54.210.196.203	TRUE	FALSE	FALSE	66	1439.340	547339401	1
0.04492	192.168.1.17	54.210.196.203	TRUE	FALSE	FALSE	66	1469.170	1783279455	1
0.06272	192.168.1.17	54.210.196.203	TRUE	FALSE	FALSE	66	1052.134	1471331497	1
0.05877	192.168.1.17	54.210.196.203	TRUE	FALSE	FALSE	66	1122.995	3083871698	1
0.05954	192.168.1.17	54.210.196.203	TRUE	FALSE	FALSE	66	1108.342	3251699352	1
0.06135	192.168.1.17	54.210.196.203	TRUE	FALSE	FALSE	66	1075.788	3976420336	1
0.07623	192.168.1.17	54.210.196.203	TRUE	FALSE	FALSE	66	865.7623	1596628946	0
0.09122	192.168.1.17	54.210.196.203	TRUE	FALSE	FALSE	66	723.5187	38244721	0
0.04415	192.168.1.17	54.210.196.203	TRUE	FALSE	FALSE	66	1494.681	186124302	1
0.06013	192.168.1.17	54.210.196.203	TRUE	FALSE	FALSE	66	1097.541	298175682	1

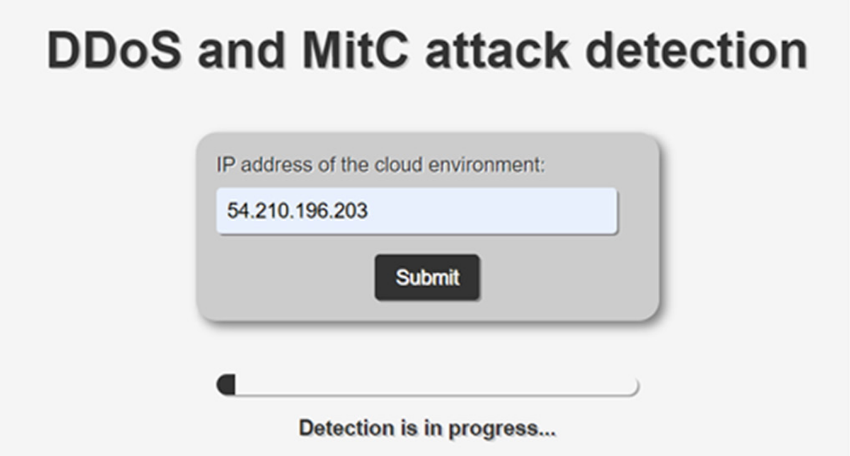
The model with the highest accuracy is then used to predict whether a DDoS or MitC attack will occur. The model is used to make a prediction for a specific event by creating an input array with six properties (*timestamp, page_id, num_requests, num_errors, duration and bytes_sent*) and feeding it to the trained model's predict() method. The result of the `predict()` method is a Boolean value indicating whether the event is likely to be a DDoS or MitC attack.

The parameters based on which the prediction of the attack time was made, are:

- **timestamp:** used to record the time at which the data was collected.
- **page_id:** a unique identifier for a particular page on a website.
- **num_requests:** the number of requests made to the page in the specified time period.
- **num_errors:** number of errors encountered while processing page requests.
- **duration:** activity time, in seconds.
- **bytes_sent:** the number of bytes sent from the server to the client during the specified time period, which represents the size of the data that was transmitted.
- **date:** a NumPy string that combines all of the above parameters into a single data point.
- **is_attack:** the predicted output of the machine learning model. It represents whether the data points are classified as an attack (1) or not (0).

4.4 Evaluation and results

To generate the results, the simulation of two attacks was done first. The execution of the program is done with IDLE Python and then it is opened in the browser as it is presented in Figure 11. In addition, a comparison of the results will be made using jMeter as a simulator of DDoS and MitC attacks. First, the IP address of the cloud environment against which the attacks will be made and then their detection is given. By providing the IP address, the port is also given (in our case 80) and the number of attacks that will be carried out (first case 100, second case 1000, third case 10000, etc.). By providing this data, the attacks begins and then the analysis of the packets explained in the sections above, and data is stored in a csv file.



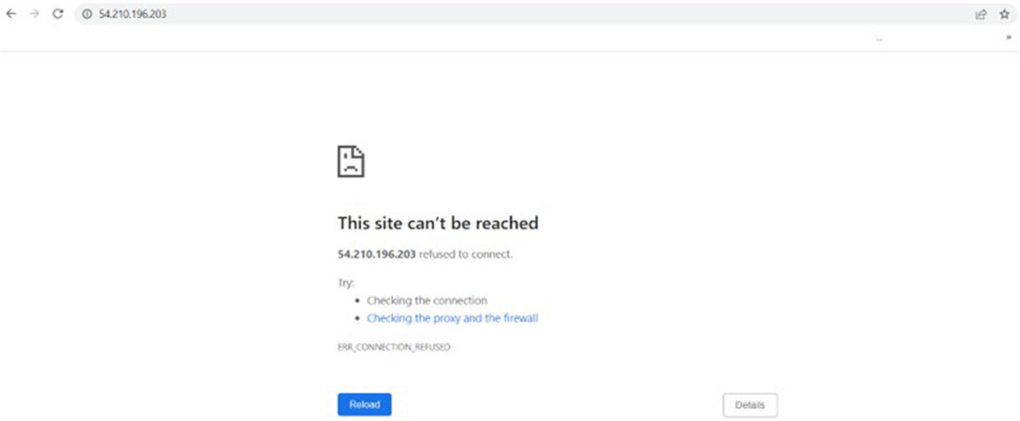
DDoS and MitC attack detection

IP address of the cloud environment: 54.210.196.203

Detection is in progress...

Fig. 11. The UI of the implemented application

In Figure 12, is presented a screen shot of the web application, after it is down (inaccessible), due to attack.



This site can't be reached

54.210.196.203 refused to connect.

Try:

- Checking the connection
- Checking the proxy and the firewall

ERR_CONNECTION_REFUSED

Fig. 12. Inaccessible application

Following the training of the four algorithms and identification of the optimal performer, the subsequent step involves utilizing this algorithm to predict the likelihood of an impending DDoS or MitC attack.

The Decision Trees algorithm outperformed all other algorithms across all test cases, as evidenced by its performance in Table 2. It presents accuracy, precision, and recall results for the scenario with a PPS threshold of 1000, clearly highlighting that the Decision Trees algorithm achieved superior outcomes compared to the other algorithms.

Table 2. Results for the four algorithms compared

Algorithm	Accuracy	Precision	Recall
Decision Trees	0.999	1.0	0.991
Naïve Bayes	0.997	0.991	0.987
SVM	0.989	0.991	0.987
KNN	0.987	0.989	0.982

Furthermore, except doing the simulation of the attacks with the application implemented, a simulation of the attacks was also done with jMeter[2]. The results obtained from the application (time duration and prediction accuracy) were compared with those obtained from jMeter.

Various parameter values were employed during the testing phase, encompassing factors such as the number of threads and diverse PPS threshold values. These specific parameters and their corresponding values were chosen to comprehensively assess the algorithm's performance under various conditions.

Table 3 shows the results obtained after simulating attacks with our application.

[2] https://jmeter.apache.org/

Table 3. Results after simulating attacks for different threshold values

Number of Threads	PPS Threshold	Time Duration	Prediction	Accuracy
100	> 1000	10 min.	False	0.1712
1000	> 50	9.5 min.	True	0.995
10000	> 100	9 min.	True	0.9981
500	> 50	11 min.	False	0.028
2500	> 10	10.5 min.	True	0.99

Table 4 shows the results obtained after simulating attacks with jMeter.

Table 4. Results after simulating attacks with jMeter for different threshold values

Number of Threads	PPS Threshold	Time Duration	Prediction	Accuracy
100	> 1000	10 min.	False	0.191
1000	> 50	9.5 min.	True	0.988
10000	> 100	9 min.	True	0.9977
500	> 50	11 min.	False	0.019
2500	> 10	10.5 min.	True	0.99

5 CONCLUSION AND FUTURE WORK

5.1 Conclusion

The detection of DDoS (Distributed Denial of Service) and MitC (Man-in-the-Cloud) attacks in a cloud environment, especially in AWS EC2 (Elastic Compute Cloud), can be achieved quite effectively through the implementation of machine learning algorithms such as Decision Trees, Naive Bayes, K-Nearest Neighbors (KNN) and Support Vector Machines (SVM). These algorithms have demonstrated their effectiveness in accurately identifying and classifying malicious activities by distinguishing them from legitimate network traffic.

Machine learning algorithms play a vital role in addressing these security challenges. By analyzing network traffic patterns, depending on user behavior, these algorithms can identify anomalous activities associated with DDoS and MitC attacks. The ability to distinguish between normal and malicious network traffic allows timely detection and mitigation of these attacks, preventing potential service interruptions, data breaches and unauthorized access. In this paper, finding the best algorithm to use for prediction had a rather large impact on the accuracy of attack detection.

Applying machine learning algorithms to detect DDoS and MitC attacks in cloud environments, such as AWS EC2, offers several advantages. These algorithms can adapt to evolving attack techniques, drawing knowledge from labeled datasets containing known attack cases, and discover previously unseen attack patterns. Their ability to handle large-scale and complex data sets makes them well-suited to the dynamic nature of cloud environments.

To wrap up, detecting DDoS and MitC attacks in a cloud environment using machine learning algorithms is essential to ensure the security, availability and confidentiality of cloud services. These attacks pose significant risks to cloud infrastructures, and effective detection and mitigation strategies are essential to guard the cloud environment. In summation, while the current study constitutes a substantial stride towards guarding the cloud, it simultaneously beckons for sustained future research efforts to harness emergent technologies, confront evolving threats, and harness the dynamic potential of machine learning.

5.2 Future work

This work carried out has room for improvements and for more extensive research, even though the results achieved in this work have reached the intended objectives and even to a very high degree. As technological advancement continues its unrelenting progression, the landscape of cyber threats too evolves in tandem, necessitating diligent endeavors to propel the current implementation towards broader horizons. The first idea for future work might include the detection of any other attacks in cloud environments. This could only be achieved by adapting the code for specific attacks. Machine learning, characterized by its intricate tapestry, remains subject to perpetual innovation and breakthroughs. Accordingly, the envisaged evolution of this field raises the prospect of even more potent algorithms, poised to yield heightened precision and efficacy in guarding the cloud. Areas of prospective development could include: the refinement of feature selection processes, the assimilation of hybrid models amalgamating diverse algorithmic paradigms, and the deepening of behavior analysis techniques.

6 REFERENCES

[1] N. Mishra, R. K. Singh, and S. K. Yadav, "Detection of DDoS vulnerability in cloud computing using the perplexed bayes classifier," *Computational Intelligence and Neuroscience*, vol. 2022, 2022. https://doi.org/10.1155/2022/9151847

[2] J. Flynn, "25 Amazing cloud adoption statistics," 2023. Online: https://www.zippia.com/advice/cloud-adoption-statistics [Accessed: February 2023].

[3] K. I. Arif and O. Nassif, "DDoS attack detection and prevention system using packet analysis and machine learning algorithms in cloud computing," *Design Engineering*, Issue 9, pp. 13782–13795, 2021.

[4] Statista, "Most common cloud security incidents worldwide in 2020 and 2022," 2023. Online: https://www.statista.com/statistics/1320178/common-cloud-securityattacks-worldwide/ [Accessed: February 2023].

[5] A. Musa, V. Vishi, and B. Rexha, "Attack analysis of face recognition authentication systems using fast gradient sign method," *Journal of Applied Artificial Intelligence*, vol. 35, no. 15, pp. 1346–1360, 2021. https://doi.org/10.1080/08839514.2021.1978149

[6] B. Rexha, G. Shala, and V. Xhafa, "Increasing trustworthiness of face authentication in mobile devices by modeling gesture behavior and location using neural networks," *Future Internet*, vol. 10, no. 2, pp. 1–17, 2018. https://doi.org/10.3390/fi10020017

[7] Rr. Thaqi, K. Vishi, and B. Rexha, "Enhancing burp suite with machine learning extension for vulnerability assessment of web applications," *Journal of Applied Security Research*, vol. 18, no. 4, pp. 789–807, 2022. https://doi.org/10.1080/19361610.2022.2096387

[8] S. Gavini, A. V. Babu, and D. Midhunchakkarvarthy, "A survey on cloud attack detection using machine learning techniques," *International Journal of Computer Applications*, vol. 175, no. 34, pp. 21–27, 2020. https://doi.org/10.5120/ijca2020920887

[9] T. Nathezhtha and V. Yaidehi, "Cloud insider attack detection using machine learning," *International Conference on Recent Trends in Advance Computing (ICRTAC)*, pp. 60–65, 2018. https://doi.org/10.1109/ICRTAC.2018.8679338

[10] V. Sharma, V. Verma, and A. Sharma, "Detection of DDoS attacks using machine learning in cloud computing," in *Advanced Informatics for Computing Research. ICAICR 2019. Communications in Computer and Information Science*, Luhach, A., Jat, D., Hawari, K., Gao, X. Z., and Lingras, P., Eds., Springer, Singapore, vol. 1076, 2019. https://doi.org/10.1007/978-981-15-0111-1_24

[11] A. Manna and M. Alkasassbeh, "Detecting network anomalies using machine learning and SNMP-MIB dataset with IP group," in *2nd International Conference on New Trends in Computing Sciences (ICTCS)*, Amman, Jordan, 2019, pp. 1–5. https://doi.org/10.1109/ICTCS.2019.8923043

[12] K. Singh, "DDOS attack detection and mitigation using statistical and machine learning methods in SDN," *National College of Ireland*, 2020. Online: https://norma.ncirl.ie/4542/1/vishalkumarsingh.pdf

[13] D. A. Varma, R. Ashish, V. Venkata Sai Sandeep, B. Venkatesh, and R. Kannadasan, "Detection of DDOS attacks using machine learning techniques: A hybrid approach," in *ICT Systems and Sustainability. Advances in Intelligent Systems and Computing*, Tuba, M., Akashe, S., and Joshi, A., Eds., Springer, Singapore, vol. 1270, 2021, pp. 439–446. https://doi.org/10.1007/978-981-15-8289-9_42

[14] M. Zekri, S. E. Kafhali, N. Aboutabit, and Y. Saadi, "DDos attack detection using machine learning techniques in cloud computing environments," in *3rd International Conference of Cloud Computing Technologies and Applications (CloudTech)*, Rabat, Morocco, 2017, pp. 1–7. https://doi.org/10.1109/CloudTech.2017.8284731

[15] A. Alshammari and A. Aldribi, "Apply machine learning techniques to detect malicious network traffic in cloud computing," *Journal of Big Data*, vol. 8, no. 90, 2021. https://doi.org/10.1186/s40537-021-00475-1

[16] A. Singh and K. Chatterjee, "Cloud security issues and challenges: A survey," *Journal of Network and Computer Applications*, vol. 79, pp. 88–115, 2017. https://doi.org/10.1016/j.jnca.2016.11.027

[17] Cloudflare, "What is a DDoS attack?" 2022. Online: https://www.cloudflare.com/learning/ddos/what-is-a-ddos-attack/ [Accessed: March 2023].

[18] B. B. Gupta, R. C. Joshi, and M. Misra, "Distributed denial of service prevention techniques," *International Journal of Computer and Electrical Engineering (IJCEE)*, vol. 2, no. 2, pp. 268–276, 2010. https://doi.org/10.7763/IJCEE.2010.V2.148

[19] A. Kahol, "Beware the man in the cloud: How to protect against a new breed of cyberattack," 2023. Online: https://www.helpnetsecurity.com/2019/01/21/mitc-attack/ [Accessed: March 2023].

[20] A. Zimba and Z. Wang, "On Man-In-The-Cloud (MITC) attacks: The analytical case of Linux," in *IEEE International Conference on Intelligence and Security Informatics (ISI)*, Beijing, China, 2017, pp. 170–172. https://doi.org/10.1109/ISI.2017.8004901

[21] Amazon Elastic Compute Cloud Documentation. Online: https://docs.aws.amazon.com/ec2/ [Accessed: August 2023].

[22] WHIZLABS, "Create Elastic Network Interface - Multiple IPs on an EC2, 2023. Online: https://www.whizlabs.com/labs/create-elastic-network-interface-multiple-ips-on-an-ec2 [Accessed: August 2023].

7 AUTHORS

Blerim Rexha is a professor at the University of Prishtina, Faculty of Electrical and Computer Engineering in Prishtina, Kosovo. He boasts of an impressive publication record in respected international conferences and journals. He's also been a guest speaker at numerous national and international conferences, highlighting

his academic and research experience including biometrics, privacy, cyber security, cryptography, and machine learning (E-mail: blerim.rexha@uni-pr.edu).

Rrezearta Thaqi is an assistant at the University of Prishtina, Faculty of Electrical and Computer Engineering, Prishtina, Kosovo. She obtained her Master's Degree from the Faculty of Electrical and Computer Engineering, University of Prishtina "Hasan Prishtina", Prishtina, Kosovo. Her research interests include Cyber Security, Cloud Security and Machine Learning (E-mail: rrezearta.thaqi@uni-pr.edu).

Artan Mazrekaj is an Assistant Professor at the University of Prishtina, Faculty of Electrical and Computer Engineering, Prishtina, Kosovo. He has published numerous papers in conferences and journals. His research interests include Cloud Computing, Distributed Systems and IoT (E-mail: artan.mazrekaj@uni-pr.edu).

Kamer Vishi holds a PhD in cybersecurity from University of Oslo, Norway, and currently he is a Senior Security Operations Manager working for the Norwegian Agency for Local Governments (KBN). His research interests include biometrics, privacy, active cyber defense solutions, post-quantum cryptography, and machine learning (E-mail: kavi@kbn.com).