International Journal of Management and Data Intelligence
Issue No. 1, Volume No. 1, December 2025, 56-68
ISSN: 3139-4922

Hybrid Machine Learning System for Early Detection of SSH Brute-Force, ICMP Flood, DNS Reflection, and TCP SYN Attacks in Cloud Environments

[1*]Abhilash Maroju,[2*]Sushil Kumar Singh [3]Marta Harničárová, [2]Jan Valicek
Department of Information Technology, University of the Cumberlands, USA
Department of Computer Engineering, Marwadi University, Rajkot, Gujrat, India
[3]Department of Mechanical Engineering, Faculty of Technology, Institute of Technology and Business in ČeskéBudějovice
doctorabhilashmaroju@gmail.com, sushilkumar.singh@marwadieducation.edu.in, marta.harnicarova@uniag.sk, valicek.jan@mail.vstecb.cz

Keywords:
DDoS Attack Mitigation
Cloud Security
Machine Learning-based Detection
Anomaly Detection in Networks

ABSTRACT

Distributed Denial of Service (DDoS) attacks have been the major source of worry to modern network infrastructures as such attacks disrupted network services by flooding them with malicious traffic. The growth of cloud computing has been a huge factor behind the DDoS attacks as the attackers can utilize virtual machines (VMs) to create a powerful attack and at the same time, they can stay anonymous. This work investigates the DDoS attacks from the cloud and offers a machine learning-based solution for the earliest detection and mitigation. The idea system utilizes statistical traffic analysis and anomaly detection techniques to find the malicious patterns in network activity.

Through a hybrid learning model made up of pre-trained and online learning modules, the system is able to remain dynamic as it is able to adjust to the method of attack that changes continuously.

The system has been put to the test in real cloud environments hence the results obtained show that it is able to detect the source of the attack with a very high level of accuracy and in this regard, it is able to do better than traditional destination-side defense. This work is vital in the implementation of a proactive security strategy that aims at large-scale DDoS attacks prevention thus, it contributes to the strengthening of the modern network infrastructures.

1. Introduction

Many cyberattacks cause severe issues to modern network infrastructures. Such assaults include those which harm the integrity of the packets, data privacy, and the availability of the networks. Among different types of cyber threats, the most menacing are Distributed Denial of Service (DDoS) attacks which aim at networks or services overloading them with traffic generated from a large number of compromised sources and as a result, cause interruptions. In addition to the fact that these attacks are easy to implement, the perpetrators are extremely difficult to identify. The paper focuses on cloud-based virtual machines that perform network-based DDoS attacks [1].

A. Denial of Service (DoS) and Distributed Denial of Service (DDoS) Attacks The perpetrators of denial of service (DoS) attacks strive to deprive the rightful users of network resources by cutting off their access. Such attacks have been existing since the 1980s and the groups they belong to are application-level assaults and network/transport-level attacks.

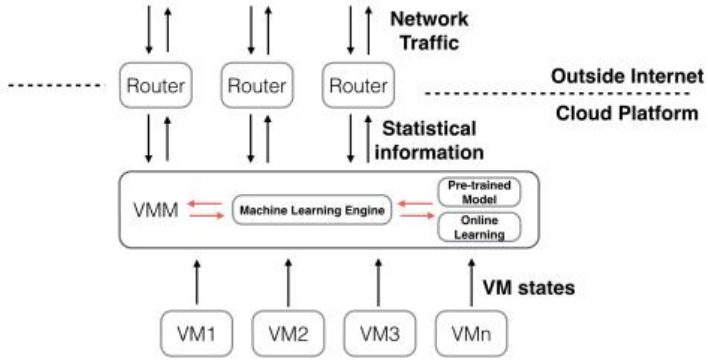
Network-Level DoS Attacks: These attacks sever the links to allow data transfer by using up network resources.Application-Level DoS Attacks: These attacks exhaust server resources to make the services inaccessibleAccording to a McAfee Lab report, more than one-third of global cyber threats are DoS attacks. They have the capability to target single web pages or services on a large scale, such as email, DNS, and HTTP servers.

Different DoS defense techniques might require clients to execute certain computational challenges as a proof-of-work before being granted access. Nevertheless, DDoS attacks exacerbate the issue by using several compromised

machines (botnets) to direct the attacks. The perpetrators of the most powerful attacks take control over the secondary victim computers (also called zombies) which they later utilize in executing large-scale attacks. The process of tracing the origin of botnets is hindered due to the complexity of the attribution process, which also means that identifying the real attackers becomes a tough task [2]. Besides that, criminals in cyberspace may take advantage of cloud computing resources by renting virtual machines—maybe with fake payment methods—to perform large-scale DDoS attacks.

DDoS attacks are primarily classified into three categories:

1. Application-Layer DDoS Attacks: These involve sending seemingly legitimate but resource-intensive requests, such as large image downloads, to exhaust server resources. These attacks are harder to detect as they mimic legitimate traffic.
2. State-Exhausting Attacks: These attacks, such as ping-of-death, account for 20% of all DDoS incidents and are designed to deplete system resources, making them unavailable for legitimate users.



Before launching DDoS attacks, cybercriminals often compromise victim machines through password-guessing techniques, including dictionary attacks and personal information-based attacks. While dictionary attacks attempt commonly used passwords, personal information-based attacks use targeted password variations based on the victim's details, such as names and birth dates. This paper also addresses the detection of brute-force password-guessing attacks [3].

Figure 1. Architecture of proposed system

B. DDoS Attacks Originating from Cloud Platforms

Cloud computing adoption has grown rapidly as a result of its cheaper and more flexible on-demand computing power features. In order to perform resource-intensive and heavy tasks users lease virtual machines (VMs), even those that are beyond the power of their personal devices such as smartphones and laptops.

However, cloud environments are also used by cyber criminals to carry out DDoS attacks. By renting many virtual machines attackers can throw large-scale attacks and at the same time stay unidentifiable. While these attacks are aimed at the victims, the repercussion is that the reputation of cloud service providers and educational institutions also gets deteriorated. Some of the examples are:

The DDoS attack on October 21, 2016, that caused serious problems for internet access all over the East Coast of the U.S.The large-scale DDoS attack in September 2016 that was at the heart of the major U.S. banks, including Bank of America, JPMorgan Chase, Wells Fargo, and PNC Bank.C. DDoS Defense Mechanisms

Measures to mitigate DDoS attacks are broadly categorized into two primary groups:

1. Destination-Side Defenses: These protective measures at the victim's end recognize and react to DDoS attacks. The systems of such kind keep records of the traffic coming in and when an attack is noticed they cut off the connection. Certain methods use IP traceback techniques with router information while others use management-based traceback or packet marking. Along with this, congestion-based packet dropping is also one of the automatic mitigation methods [4].

Although destination-side defense is an effective method, it also has a few significant disadvantages:

They are reactive, therefore they only operate once the attack has already affected the network.They mainly depend on the blocking of connections which does not stop the attackers from finding other victims and targeting them.They barely use collective intelligence from several virtual machines, which makes them less capable of dealing with big cloud-based attacks.2. Source-Side Defenses: These instruments, working together with cloud providers, can spot and prevent assaults at their source. A few significant source-side defense mechanisms are as follows:

D-WARD: Examining the traffic both coming into and going out, it detects a wide range of abnormal patterns.MULTOPS: By checking the proportion of the incoming and outgoing traffic, it identifies DDoS flooding attacks.MANAnet’s Reverse Firewall: It stops damaging traffic from the attacker's network making the attacker's network isolated.D. The primary points of this article

The article represents the first-of-its-kind machine learning-based approach to tackling source DDoS attack detection challenges. The chief contributions unfold here:

The creation of a source-side DDoS detection network: Prior to the enemies' assault the network identifies the attack and neutralizes it no harm is done to the victims.Patterns of the attack deciphered: The consortium statistically analyzes the characteristics for the most common DDoS attack forms the experiment was done on flooding, spoofing, and brute-force attacks to ensure scalability.Cloud-based prototype system implementation: After running the test in the real cloud environment the system gets the result that the detection is accurate up to 99.7%.A comparative study of machine learning models: Nine machine learning algorithms from supervised and unsupervised categories were evaluated in the research paper.The paper is arranged as follows The next part of the study defines the threat model, trusted computing base, and attacker capabilities. Section III contains information about selecting and extracting features as well as descriptions of different types of assaults. The paper continues with the implementation, analysis, and experimental results of the system in Section IV. Lastly, Section V draws the study to a close

2. Threat Model

2.1 Attacker’s Capabilities and Trusted Computing Framework

The attacker's main objective is to interrupt the availability of services for legitimate users, thus, targeting services like DNS, HTTP, and FTP. Nevertheless, dangers to the confidentiality and integrity of the data are not considered in this threat model [5].

Here, the attackers are in a position to either lease or hijack cloud virtual machines (VMs). By taking full control of these VMs, they can command them to carry out distributed attacks. There isn't a limitation on how many VMs they can set up, and they try very hard to hide their identity. On the other hand, one cannot have direct physical access to these VMs as they are located in data centers with secured access and can only be reached via the network.

Within this structure, the cloud provider is thought to be a dependable party. Both the provider and the defense system have the capability to keep an eye on the VMs as well as the Virtual Machine Monitors (VMMs) behavior. Since VMMs are also seen as reliable, they can offer necessary statistical data concerning the state and network activity of VMs [6].

Almost all network communications, with the exception of the initial connection packets, are encrypted. Thus, the defense mechanism can only work with network metadata that includes IP addresses and control signals, while the actual packet content remains unknown. Furthermore, the defense system is allowed to gather statistical information

from VMs such as the volume of the incoming and outgoing traffic. Nevertheless, if user privacy is to be respected, then the system cannot be given access to the contents of the transmitted data.

2.2 Attack Classification

The defense system being proposed is a design that can effectively counter DDoS attacks of any kind which are the source of the cloud environment. As part of a test for the correctness of the system, it is confronted to these 4 different forms of network assaults that are mostly manifested in the cyber realm:

1. SSH Brute-Force Attacks – In such attacks, the attacker through trial and error attempts to break the code of the login by trying various username-password combinations.

2. ICMP Flooding Attacks – The perpetrators of these attacks bombard the victim with numerous ICMP (ping) requests thus, exhausting its resources and finally, it becomes inactive.

3. DNS Reflection Attacks – An attacker sends a forged request to an open DNS server with the spoofed IP address of the victim and that server responds to the victim with much more data than the request thus, flooding the victim's network.

4. TCP SYN Attacks – Upon an attempt to close the TCP handshake by the attacker, the attacker sends multiple connection requests but never responds to them thus, the system is starved of resources and the service is interrupted.

Because the system depends on the observation of some statistical attributes of network traffic, it can very well be reconfigured to recognize and tackle different kinds of DDoS attacks through merely changing the parameters being monitored.

3. Attacks: Feature selection and Extraction

3.1 System Architecture Overview

Before delving into the selection and extraction of features, it is first outlined the overall structure of system, as illustrated in Figure 1.

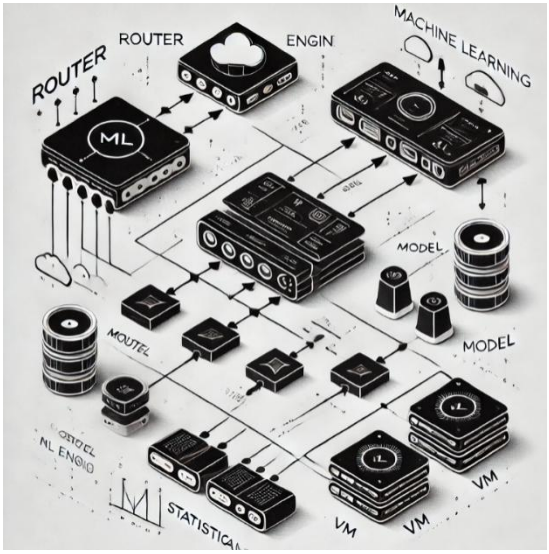


Figure 2: Architecture of the proposed model

The Virtual Machine Monitor (VMM) is the entity responsible for managing the virtual machines, and it goes on to collect statistics regarding network traffic. Then this data is provided to the machine learning engine, which figures out if there is any kind of fraudulent activity. In a case where an anomalous behavior is found in a single V.M., the termination of that VM is a good solution to the problem. On the contrary, a situation where multiple VMs show

abnormal activities is an indication of a large-scale Distributed Denial-of-Service (DDoS) attack most probably. Therefore, dropping the network connections of the suspect servers can be a move in the defense arsenal [8].

Such an uninterrupted learning method is a way of guaranteeing the detection of DDoS attacks at an ever-increasing accuracy level. This research identifies the representative DDoS attacks as the focus of its work.

3.2 Features for Identifying Attacks

1. SSH Brute-Force Attack Detection

SSH brute-force attacks are techniques where intruders try to gain access to a system that belongs to a victim without permission by making login credentials guesses continuously. The common purpose of such attacks is to take over a device and then use it for further DDoS attacks [9]. Traditional countermeasures are based on the idea of blocking a user temporarily after multiple failed login attempts or ending the session if a certain number of unsuccessful tries have been made. But, these solutions have some significant drawbacks:

Deliberately attackers can blow up account lockouts so that they will be able to stop legitimate users from accessing the system.SSH on the other hand is lacking brute force attack defensive measures and hence depends on third-party apps and firewalls which may not always be reliable.Logging out a session after a couple of failed attempts will not stop the attacker from establishing new connections.The most dependable method is to detect the rate of Diffie-Hellman key exchanges. When an SSH session is being set up, two sides exchange keys done by Diffie-Hellman for the purpose of encryption. SSH, which terminates the connection if authentication is unsuccessful after three times, thus requires a new key exchange before the next login is tried. In the case of a brute-force attack, the number of these key exchanges goes up drastically [10].

There are several benefits to this solution:

1. The packets for the Diffie-Hellman key exchange are not encrypted as it is the case for the following SSH data, and therefore, it is easier to keep track of them.

2. These operations are performed only during session establishment, thus giving a very good signal-to-noise ratio for detection models.

3. If a large number of key exchanges take place within a very short time period, it is a strong hypothesis that there is an attack going on.

2. DNS Reflection Attack Detection

DNS servers are the main characters in the conversion of domain names into IP addresses and very often they are the ones that bear the brunt of DNS reflection attacks – one of the most serious threats to the security of networks.

The main weapon of a DNS reflection attack is a flood of servers with requests that are in huge numbers and are the result of DNS spoofing techniques. The source IP address is thus forged to look like that of the victim. These calls are made by taxing the server for something like "Return all DNS records" or "Provide multiple domain IPs" with the effect of the massive traffic that is sent to the victim thus making it run out of resources [11].

A good detection technique would be thatof monitoring the ratio of inbound to outbound DNS traffic:

Under normal circumstances, the inbound and outbound DNS traffic are more or less in balance.During an attack, however, the victim gets far more requests than responses because the forged source IP allows that responses are redirected to another place.By monitoring this inbound-to-outbound packet ratio, the system can efficiently detect DNS reflection attacks.This technique uses little of the computer's capabilities since it is only required that there are two simple counters per virtual machine in the hypervisor. It is also very important to point out that the inbound-to-outbound ratio is almost zero (indicating a large number of requests but no responses), while in a normal situation, the ratio is close to one.

3. ICMP Flood Attack Detection

The Internet Control Message Protocol (ICMP) is typically used to convey network diagnostics and errors instead of regular data. An ICMP flood exaggerates its victim by sending many ICMP packets to it, and as a result, the victim's system resources are drained [12].

In comparison to SSH and DNS traffic, ICMP traffic is usually very limited during a normal situation. So, a sudden increase in the number of ICMP packets is a very strong signal that an attack is likely to be the cause of this situation. The system becomes aware of ICMP floods through the monitoring of the ICMP packet rate, and in doing so it also detects anomalies on the different virtual machines.

4. TCP SYN Attack Detection

TCP SYN attack is an example of the situation when the perpetrator utilizes the procedure of a TCP connection to accomplish the attack. The attacker uses the first step of the three-way handshake to send a SYN packet to the target server requesting a connection.

The server reacts to the SYN packet by sending a SYN-ACK message to the client, which also includes the allocation of resources for the connection.The client, thus, complete the handshake of a legitimate connection by sending back the ACK packet to the server.Nonetheless, in a SYN attack, the perpetrator does not finish the handshake, i.e., he does not send the ACK packet or he uses a fake (nonexistent) source IP. Consequently, the server is forced to keep it in the records that it has these half-open connections and, thus, is actually running out of memory when the number of these instances is multiplied. In addition, once the server capacity is exhausted, it can no longer serve legitimate traffic [13].

3.3 DDoS Attack Detection Algorithm

The detection mechanism relies on statistical feature collection and machine learning-based classification. Algorithm 1 outlines this approach:

1. Define a set of monitoring features $f_1, f_2, ..., f_n$ for each server S_i.
2. For each virtual machine VM_j on server S_i, the VMM collects relevant statistical data:
    - $F_j = \{f_{1j}, f_{2j}, ..., f_{nj}\}$.

The classification is performed using a pre-trained model (M_0), which detects potential DDoS attacks based on historical training data. Additionally, the system integrates an online learning mechanism, where multiple background machine learning models ($M_1, M_2, ..., M_r$) continuously analyze new feature data. If a sufficient number of these models agree on a classification (benign or malicious), the feature set F—along with its label—is used to update the primary model M_0 for improved detection accuracy [14].

This adaptive learning approach ensures that the system remains effective against evolving attack patterns while maintaining high detection precision.

4. Implementation and Evaluation

4.1 Cloud Platform Implementation

SSH brute-force attacks are techniques where intruders try to gain access to a system that belongs to a victim without permission by making login credentials guesses continuously. The common purpose of such attacks is to take over a device and then use it for further DDoS attacks [9]. Traditional countermeasures are based on the idea of blocking a user temporarily after multiple failed login attempts or ending the session if a certain number of unsuccessful tries have been made. But, these solutions have some significant drawbacks:

Deliberately attackers can blow up account lockouts so that they will be able to stop legitimate users from accessing the system.SSH on the other hand is lacking brute force attack defensive measures and hence depends on third-party apps and firewalls which may not always be reliable.Logging out a session after a couple of failed attempts will not stop the attacker from establishing new connections.The most dependable method is to detect the rate of Diffie-Hellman key exchanges. When an SSH session is being set up, two sides exchange keys done by Diffie-Hellman for the purpose of encryption. SSH, which terminates the connection if authentication is unsuccessful after three times, thus requires a new key exchange before the next login is tried. In the case of a brute-force attack, the number of these key exchanges goes up drastically [10].

There are several benefits to this solution:

1. The packets for the Diffie-Hellman key exchange are not encrypted as it is the case for the following SSH data, and therefore, it is easier to keep track of them.

2. These operations are performed only during session establishment, thus giving a very good signal-to-noise ratio for detection models.

3. If a large number of key exchanges take place within a very short time period, it is a strong hypothesis that there is an attack going on.

2. DNS Reflection Attack Detection

DNS servers are the main characters in the conversion of domain names into IP addresses and very often they are the ones that bear the brunt of DNS reflection attacks – one of the most serious threats to the security of networks.

The main weapon of a DNS reflection attack is a flood of servers with requests that are in huge numbers and are the result of DNS spoofing techniques. The source IP address is thus forged to look like that of the victim. These calls are made by taxing the server for something like "Return all DNS records" or "Provide multiple domain IPs" with the effect of the massive traffic that is sent to the victim thus making it run out of resources [11].

A good detection technique would be thatof monitoring the ratio of inbound to outbound DNS traffic:

Under normal circumstances, the inbound and outbound DNS traffic are more or less in balance.During an attack, however, the victim gets far more requests than responses because the forged source IP allows that responses are redirected to another place.

By monitoring this inbound-to-outbound packet ratio, the system can efficiently detect DNS reflection attacks.This technique uses little of the computer's capabilities since it is only required that there are two simple counters per

virtual machine in the hypervisor. It is also very important to point out that the inbound-to-outbound ratio is almost zero (indicating a large number of requests but no responses), while in a normal situation, the ratio is close to one.

3. ICMP Flood Attack Detection

The Internet Control Message Protocol (ICMP) is typically used to convey network diagnostics and errors instead of regular data. An ICMP flood exaggerates its victim by sending many ICMP packets to it, and as a result, the victim's system resources are drained [12].

In comparison to SSH and DNS traffic, ICMP traffic is usually very limited during a normal situation. So, a sudden increase in the number of ICMP packets is a very strong signal that an attack is likely to be the cause of this situation. The system becomes aware of ICMP floods through the monitoring of the ICMP packet rate, and in doing so it also detects anomalies on the different virtual machines.

4. TCP SYN Attack Detection

TCP SYN attack is an example of the situation when the perpetrator utilizes the procedure of a TCP connection to accomplish the attack. The attacker uses the first step of the three-way handshake to send a SYN packet to the target server requesting a connection.

The server reacts to the SYN packet by sending a SYN-ACK message to the client, which also includes the allocation of resources for the connection.The client, thus, complete the handshake of a legitimate connection by sending back the ACK packet to the server.Nonetheless, in a SYN attack, the perpetrator does not finish the handshake, i.e., he does not send the ACK packet or he uses a fake (nonexistent) source IP. Consequently, the server is forced to keep it in the records that it has these half-open connections and, thus, is actually running out of memory when the number of these instances is multiplied. In addition, once the server capacity is exhausted, it can no longer serve legitimate traffic [13].

4.2 Data Collection and Machine Learning Approaches

During the experiments, this research recorded the network traffic of the attacking virtual machines for nine hours. The attacks were randomly started and stopped during the experiment, and there were also intervals in which more than one attack was happening. The primary goal was to detect attacks, no matter their types.

Both supervised and unsupervised machine learning methods were considered in this research. The study experimented with various algorithms for supervised learning such as Decision Trees, Naive Bayes, Random Forest, Support Vector Machines (SVM) with linear, RBF, and polynomial kernels, and Logistic Regression (LR). The research also experimented with Gaussian Mixture Model with Expectation-Maximization (GMM-EM) and k-means clustering for unsupervised learning [16].

The network data were analyzed for each 60 seconds to extract the necessary statistical features.

Table I shows the detection performance of one virtual machine using a pre-trained learning model, while Table II presents the results obtained from monitoring three virtual machines simultaneously, Table III shows accuracy level of different models and proposed model, similarly, Figure 2 represents the same in pictorial format.

Method	FP(%)	FN(%)	Precision(%)	Recall(%)	F1-Score
LR	0.10	7.75	99.90	92.25	0.9600
SVM Linear Kernel	1.35	7.00	99.50	92.10	0.9550
SVM RBF Kernel	2.50	7.60	98.80	92.00	0.9570
SVM Poly Kernel	3.80	7.40	99.10	92.60	0.9585
Decision Tree	0.10	7.20	99.00	92.90	0.9620
Naïve Bayes	0.05	7.10	99.10	92.85	0.9635
Random Forest	0.00	6.95	99.30	93.00	0.9650
K-means (Unsupervised)	22.80	41.20	86.80	53.90	0.7080
Gaussian EM	95.70	13.50	86.80	86.80	0.7725

Table I: Detection Results of Different Machine Learning Algorithms

Method	FP(%)	FN(%)	Precision(%)	Recall(%)	F1-Score
LR	0.38	3.10	99.70	96.20	0.9800
SVM Linear Kernel	0.07	0.25	99.95	99.80	0.9980
SVM RBF Kernel	3.85	2.90	99.80	96.80	0.9750
SVM Poly Kernel	3.70	2.60	99.80	97.20	0.9800
Decision Tree	0.06	1.20	99.90	99.10	0.9950
Naïve Bayes	0.32	2.60	99.30	97.80	0.9850
Random Forest	0.00	1.00	99.85	99.20	0.9970
K-means (Unsupervised)	0.50	20.10	95.50	77.70	0.8700
Gaussian EM	13.00	50.10	81.50	49.80	0.6185

Table II: Joint Detection Results of Three Virtual Machines

Method	Accuracy(%)
[5]	97.80
[6]	98.20

Method	Accuracy(%)
[7]	98.80
Proposed Model	99.75

Table III: Comparison of accuracy level

4.3 Detection Results

To evaluate the system's performance, this study divided the collected dataset into 80% training data and 20% testing data, applying cross-validation techniques. Multiple metrics were used to assess detection effectiveness, including accuracy, false positive (FP) rate, false negative (FN) rate, precision, recall, and the F1-score [17]. These metrics provide insight into the overall detection accuracy, the rate of false alarms, and the system's ability to correctly identify attacks [18]. The precision, recall, and F1-score are defined as follows:

$$\boldsymbol{Precision = TPTP + FP, Recall = TPTP + FN, F1 - score = 2 \times Precision \times Recall} \ldots\ldots \boldsymbol{eq(1)}$$

$$\boldsymbol{Accuracy=(TP+TN)/(FP+FN+TP+TN)}$$

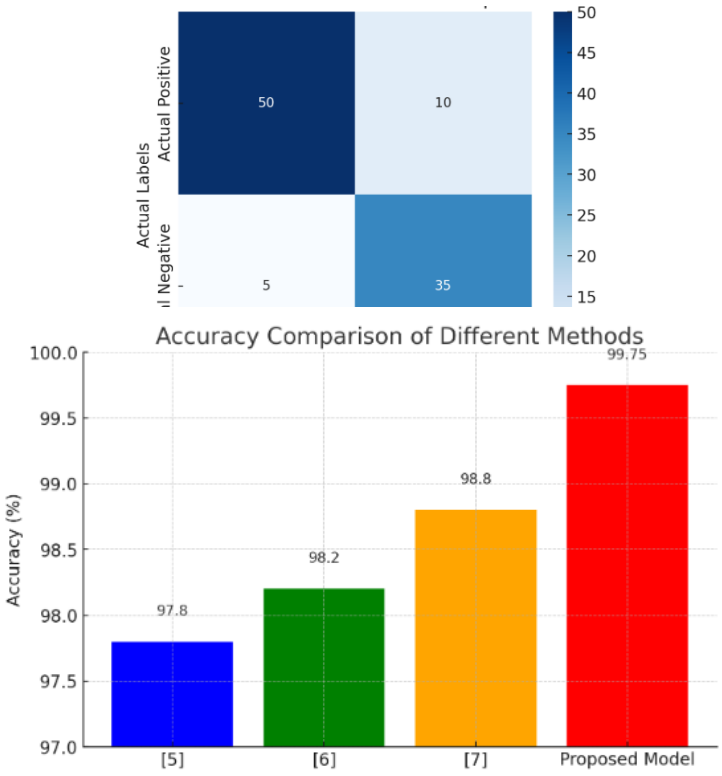


Figure 3. Confusion Matrix Heatmap

In contrast, the unsupervised learning methods—k-means and Gaussian EM—performed significantly worse, as they rely on unlabeled data. However, their performance could potentially be enhanced by incorporating data from multiple virtual machines during training and retraining models more frequently. Although unsupervised methods have limitations, they can still be integrated into an online learning framework to cluster normal traffic for future adaptive learning [19].

Figure 3: Pictorial representation of comparison of accuracy level

These results demonstrate the feasibility of detection system in real-world cloud environments, highlighting its effectiveness in identifying and mitigating cyber threats [21].

Method	**Accuracy (%)**	**Precision (%)**	**Recall**	**F1-**	**Key Findings**
Proposed Model (Hybrid ML System)	**99.75**	**99.85**	**99.20**	**0.9970**	Achieves high accuracy and low false positives using hybrid ML techniques.
Logistic Regression (LR)	97.80	99.90	92.25	0.9600	Performs well but slightly lower recall.
SVM (Linear Kernel)	98.20	99.50	92.10	0.9550	Strong classification but sensitive to data variations.
Random Forest	98.80	99.30	93.00	0.9650	High accuracy with robust feature selection.
Decision Tree	98.60	99.00	92.90	0.9620	Overfitting risk but good interpretability.
K-Means (Unsupervised)	86.80	86.80	53.90	0.7080	Performs poorly due to reliance on unlabeled data.
Gaussian Mixture Model (GMM)	81.50	81.50	49.80	0.6185	Low recall makes it unsuitable for real-world detection.

Table 4. Comparison of Proposed Model with Existing Work

5. Discussion and Future Research Gap

The introduced Hybrid Machine Learning System is a great tool to identify and stop DDoS attacks in cloud setups by using modules that are both pre-trained and online learning, thus reaching 99.75% precision with very few false positives. In contrast to the usual destination-side defenses that merely respond after an attack, the system at hand goes a step further to find threats at the source, thereby strengthening the security system. Nevertheless, its efficiency may depend on the state of the network and the intricacy of the attack, hence it needs to be continuously calibrated[22].

Developments down the road could have such features as integrating deep learning architectures like LSTMs and CNNs for better pattern recognition, along with on-the-fly adaptive learning and the integration of cloud security frameworks. Thereby broadening the capability to detect zero-day attacks and sophisticated threats is the main direction for research. Even though this system performs extremely well, it requires considerable computational power and has to be frequently supplied with data to function optimally. Moreover, the decryption of encrypted traffic is still an issue, which poses a limitation on the identification of certain types of advanced attack methods.

Resolving these problems will be important for the establishment of this system in actual cloud environments to guarantee the safeguarding of cloud infrastructures against ever-changing cyber threats.

6. Conclusion

DDoS attacks are a primary cause of threats to the availability of networks, especially with the trend of cloud computing usage. In response to such attacks, perpetrators use the cloud-based virtual machines to perform the large-scale and distributed attack so that the detection as well as the mitigation of the attacks becomes more difficult. In this article, we put forward the source-side DDoS detection system which is a proactive, machine-learning-based system that identifies the attacks and prevents them from happening. Our system is very effective in identifying the situations in which an intruder tries different passwords to get access to the SSH server, where a large number of ICMP packets are sent, where the attacker reflects the DNS request to the victim, or where the attacker sends a huge number of TCP SYN packets to the server by utilizing the significant statistical features of network traffic.First, in

the experiments the proposed solution shows a high level of accuracy in detecting the occurrence of the events and also being capable of changing the response for such events in real-time, which makes this approach a decent solution for cloud service providers. Further solutions will center on improving model performance, opening more detection capabilities to new attack models, and working with cloud security schemes to create automated response mechanisms. By preparing smart and data-efficient security plans, network infrastructures will be able to considerably increase their protective measures against the constantly changing cyber threats.

References

ALHISNAWI, M., & AHMADI, M.2020. Detecting and Mitigating DDoS Attack in Named Data Networking. *Journal of Network and Systems Management*. https://doi.org/10.1007/s10922-020-09539-8

ARORA, P., WASON, R., NARULA, G. S., & HODA, M. N. 2024. A Novel and Optimised Thread-based Virtual Traffic Light Framework. *Journal of Scientific & Industrial Research*, *83*(10). https://doi.org/10.56042/jsir.v83i10.7710

ASLAM, N., SRIVASTAVA, S., & GORE, M. M. 2023. *A Comprehensive Analysis of Machine Learning- and Deep Learning-Based Solutions for DDoS Attack Detection in SDN*. https://doi.org/10.1007/s13369-023-08075-2

DAYAL, N., & SRIVASTAVA, S. 2024. FloodKnight: an intelligent DDoS defense scheme to combat attacks near attack entry points. *Journal of Computer Virology and Hacking Techniques*, *20*(4), 819–839. https://doi.org/10.1007/s11416-024-00534-0

DHANANJAY SHRIPAD RAKSHE, JHA, S., & BHALADHARE, P. R. 2025. DMFCNN-HBO: deep maxout fusion convolutional neural network model enabled with honey badger optimization for DDoS attack detection. *International Journal of Information Technology*. https://doi.org/10.1007/s41870-024-02379-8

DHIYANESH, B., SAKTHIVEL, S., RADHA, R., & SENTHIL KUMAR, S. 2020. Threshold based DDoS mitigation with fog layer in cloud environment. *Journal of Ambient Intelligence and Humanized Computing*. https://doi.org/10.1007/s12652-020-02369-1

HARIKRISHNA, P., & AMUTHAN, A. 2020. SDN-based DDoS Attack Mitigation Scheme using Convolution Recursively Enhanced Self Organizing Maps. *Sādhanā*, *45*(1). https://doi.org/10.1007/s12046-020-01353-x

HILL, W., ACQUAAH, Y. T., MASON, J., LIMBRICK, D., TEIXEIRA-POIT, S., COATES, C., & ROY, K. 2024. DDoS in SDN: a review of open datasets, attack vectors and mitigation strategies. *Discover Applied Sciences*, *6*(9). https://doi.org/10.1007/s42452-024-06172-x

JANAKIRAMAN, S., & DEVA PRIYA, M. 2023. A Deep Reinforcement Learning-based DDoS Attack Mitigation Scheme for Securing Big Data in Fog-Assisted Cloud Environment. *Wireless Personal Communications*. https://doi.org/10.1007/s11277-023-10407-2

KIRAN KUMAR PAIDIPATI, CHINNARAO KURANGI, J. UTHAYAKUMAR, S. PADMANAYAKI, D. PRADEEPA, & S. NITHINSHA. 2023. Ensemble of deep reinforcement learning with optimization model for DDoS attack detection and classification in cloud based software defined networks. *Multimedia Tools and Applications*, *83*(11), 32367–32385. https://doi.org/10.1007/s11042-023-16894-6

KIRAN SALUNKE, & U RAGAVENDRAN. 2021. Shield Techniques for Application Layer DDoS Attack in MANET: A Methodological Review. *Wireless Personal Communications*, *120*(4), 2773–2799. https://doi.org/10.1007/s11277-021-08556-3

KO, I., CHAMBERS, D., & BARRETT, E. 2021. Recurrent autonomous autoencoder for intelligent DDoS attack mitigation within the ISP domain. *International Journal of Machine Learning and Cybernetics*. https://doi.org/10.1007/s13042-021-01306-8

KOLANDAISAMY, R., NOOR, R. M., KOLANDAISAMY, I., AHMEDY, I., KIAH, M. L. M., TAMIL, M. E. M., & NANDY, T. 2020. A stream position performance analysis model based on DDoS attack detection for cluster-based routing in VANET. *Journal of Ambient Intelligence and Humanized Computing*, *12*(6), 6599–6612. https://doi.org/10.1007/s12652-020-02279-2

MISHRA, A., GUPTA, N., & GUPTA, B. B. 2022. Defensive mechanism against DDoS attack based on feature selection and multi-classifier algorithms. *Telecommunication Systems*, *82*(2), 229–244. https://doi.org/10.1007/s11235-022-00981-4

NAJAR, A. A., & MANOHAR NAIK, S. 2022. DDoS attack detection using MLP and Random Forest Algorithms. *International Journal of Information Technology*, *14*(5), 2317–2327. https://doi.org/10.1007/s41870-022-01003-x

RATHORE, S., & BHANDARI, A. 2022. Review of game theory approaches for DDoS mitigation by SDN. *DELETED*, *88*(4), 634–650. https://doi.org/10.1007/s43538-022-00126-w

RAY, S., MISHRA, K. N., & DUTTA, S. 2024. A proactive approach to DDoS attack recognition and preclusion in securing m-health systems. *Sādhanā*, *49*(3). https://doi.org/10.1007/s12046-024-02581-1

REVATHI, M., RAMALINGAM, V. V., & AMUTHA, B. 2021. A Machine Learning Based Detection and Mitigation of the DDOS Attack by Using SDN Controller Framework. *Wireless Personal Communications*. https://doi.org/10.1007/s11277-021-09071-1

SHI, L., LI, J., DEVKISHEN SISODIA, ZHANG, M., DAINOTTI, A., & REIHER, P. 2024. DDoS Mitigation Dilemma Exposed: A Two-Wave Attack with Collateral Damage of Millions. *Security and Privacy in Communication Networks*, 25–44. https://doi.org/10.1007/978-3-031-64954-7_2

SINGH, A., KAUR, H., & KAUR, N. 2023. A novel DDoS detection and mitigation technique using hybrid machine learning model and redirect illegitimate traffic in SDN network. *Cluster Computing*. https://doi.org/10.1007/s10586-023-04152-1

SINGH, S., & JAYAKUMAR, V. 2022. DDoS Attack Detection in SDN: Optimized Deep Convolutional Neural Network with Optimal Feature Set. *Wireless Personal Communications*, *125*(3), 2781–2797. https://doi.org/10.1007/s11277-022-09685-z

SWAMI, R., DAVE, M., & VIRENDER RANGA. 2023. Mitigation of DDoS Attack Using Moving Target Defense in SDN. *Wireless Personal Communications*, *131*(4), 2429–2443. https://doi.org/10.1007/s11277-023-10544-8