Cyber Attack Detection Using Bellman Optimality Equation in Reinforcement Learning

Monali Shetty[1(✉)] and Sharvari Tamane[2]

[1] MGM University, Aurangabad, India
shettymonalin@gmail.com
[2] UDICT, MGM University, Aurangabad, India
hoditjnec@mgmu.ac.in

Abstract. Cyber-attacks have been always a part of any application that is run on internet. It is a quintessential need to protect vulnerable data from being attacked by malicious people. Along with that it is also necessary to prevent any mishap that has been planned by hackers to destroy or tamper a company's progress or ongoing work. This research gives a solution that uses reinforcement learning to understand and learn the attack beforehand so that whenever the attack is performed in the future, it will be caught by the system and the authorities can perform preventative measures to avoid the damage. The main objective of this research is dealing with Denial of Service attack in network. The research provides an in-depth view of how attacks can be manifested by malicious individuals. It is also to be noted that the simulation of attacks has also been carried out to understand it better and provide solution for the same in the form of detection mechanism. The use of artificial intelligence, reinforcement learning provides the agent a freedom to explore the environment and understand what an attack is and what is not unlike traditional machine learning algorithms that are restricted to a particular dataset which requires a lot of maintenance and storage.

Keywords: cyber-attack · detection · reinforcement learning · DDOS

1 Introduction

This paper will discuss some related study on the detection of cyber-attacks. A distributed denial of service (DDoS) attack is the most dangerous in terms of network security. It is now challenging to identify these attacks and protect internet services from them. The suggested solution employs a machine learning-based methodology to identify and categorize various network traffic flow patterns.

1.1 Cybersecurity

The practice of preventing malicious attacks on data, mobile devices, networks, electronic systems, and computers is known as cybersecurity. It is frequently called "electronic data security" or "cybersecurity in information technology". Network security, information security, and operational security are the three main subcategories

S. Tamane et al. (Eds.): ICAMIDA 2022, ACSR 105, pp. 188–200, 2023.
https://doi.org/10.2991/978-94-6463-136-4_19

of the term. Common Attacks are malware, SQL injection, man-in-the-middle attack, distributed denial-of-service attack (DDOS).

1.2 Reinforcement Learning

Machine Learning includes reinforcement learning. It all comes down to taking the proper procedures to maximize your return in a specific situation. It is well-used by many software programs and computers to decide what the finest course of action is to take in a particular situation. In contrast to supervised learning which also includes the answer key so that the model may be taught through the right response, reinforcement learning does not include the answer and instead depends on the reinforcement agent to determine how to carry out the task. Without a training dataset, it is forced to draw lessons from its own past experiences.

2 Background

Consider the following scenario: you are viewing several websites, one of which appears to be a bit sluggish. You may fault their servers for needing to increase its scalability, since they may be facing a high volume of users visiting their website. Most sites have previously considered this topic. They may have been the victim of a DDoS assault, or Distributed Denial of Service attack. The goal of conventional Distributed Denial of Service (DDoS) attacks is to take advantage of system bottlenecks. This is done by flooding an application with more requests than it can process, sending it even more traffic than its network card can handle, etc. Traditional DDoS attacks target a system's one fixed point of failure, like a web server. An organization's website might not be accessible to visitors if the web server is down.

2.1 DDoS Attack

A DDoS attack involves an attacker trying to block access to a specific service by redirecting copious volumes of traffic from numerous end systems. Due to the high volume of traffic, network resources are devoted to responding to requests from those fake end systems, blocking legitimate users from using the resources.

DDoS attack types — there are three different types for DDoS attacks: Attacks against the application layer are aimed at layer 7, which is where websites are created in response to user requests, in the OSI model. Examples include HTTP Flood and attacks on DNS services.

A sort of cyber-attack is a protocol attack, commonly referred to as a state-exhaustion attack. These attacks target holes in layers 3 and 4 of the protocol stack. Ping of Death and the SYN Flood attack are two instances. And third type is volumetric attacks: To render a network unusable for users, volumetric attacks overwhelm it through amplification or a botnet. They are simple to create by just sending a large amount of traffic to the desired server. Instances include DNS and NTP amplification as well as TCP and UDP flood attacks. DDoS attacks are frequently used, and they include: SYN Flood attack (Fig. 1).

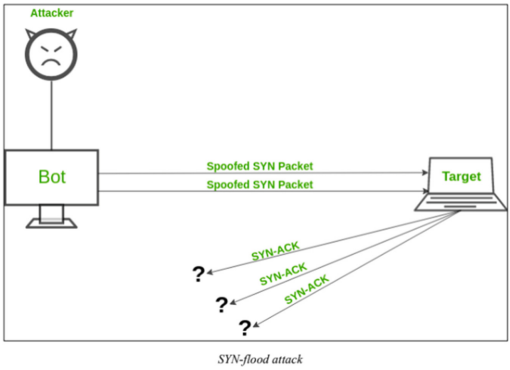

Fig. 1. SYN-flood attack

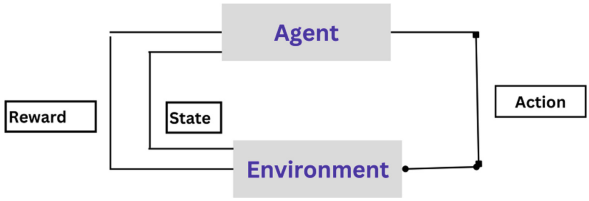

Fig. 2. Q-learning model

The mechanism of a SYN Flood attack is similar to a mischievous child that keeps knocking on the door (request) and escaping. When the elderly person exits and opens the door, nobody is there to respond. The elderly person eventually grows jaded after experiencing numerous instances of the same things and stops responding to anyone, not even actual people. A SYN attack exploits the TCP Handshake by releasing SYN packets with a false IP address. However, the vulnerable server does not receive a final acknowledgement and keeps replying.

2.2 Q-Learning Model

A reinforcement learning approach called Q-Learning will decide which line of action is optimal based on the present situation. It opt for this activity at random with the objective of raising the reward (Fig. 2).

An off-policy, model-free reinforcement learning algorithm called Q-learning will decide the agent's best line of action given its present situation. The subsequent action will be conducted in accordance with the agent's location within the environment. Determining the best course of action is the model's objective in the current situation. It may do this by establishing its own rules or by acting inconsistently with the designated policy. This is referred to as being "off-policy" because it implies that a policy is not truly required. Model-free means that the agent moves forward using predictions of how the environment will respond. Instead of a reward system, it uses trial and error to learn. Important terms for the Q-Learning are the State S, represents the agent's position at any

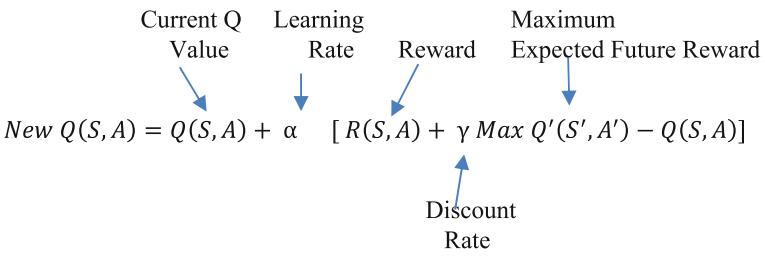

Fig. 3. Bellman Equation

given time in an environment. The action A, the agent performs action A when a specific situation exists. The agent will either get a good reward or a bad reward for each activity. Episodes happen whenever an agent is unable to start a new action because they are in a terminating circumstance. Q-Values: Used to gauge the effectiveness of an action conducted at a specific condition (A, S). Temporal Difference: A formula that compares the values of the current and earlier states and activities to determine the Q-Value. What Is the Bellman Equation, Exactly? To estimate the value of a specific circumstance, one uses the Bellman Equation as well as the benefits of being in or accepting that condition. We will receive the best value in the perfect state. The answer is displayed below. The future state of our agent is predicted using the current state, the reward connected to that state, the maximum expected reward, and a discount rate that reflects how much the reward would be worth in the current state. The model's learning rate determines how quickly and otherwise slowly it will learn. Figure 3 shows Bellman Equation.

A Q-Table's Creation: When our algorithm is run, we will run into a number of different answers, and thus the agent will take a number of different paths. Who among them is the best, and how can we tell? We achieve this by compiling our results into a table known as a Q-Table. We could decide what's best to do for each environmental state using a Q-Table. The desired state and reward in the future are determined at each state using the Bellman Equation, and they are then recorded in a table for comparison with other states. Constructing a q-table for a command-driven agent who should learn how to run, retrieve, and sit. The steps for creating a q-table are as follows: Create a Q-Table first, containing 0s for all of the values. All states and prizes will start out with values of zero.

3 Literature Review

A new dataset that includes a mix of many attacks of the current day, including SID DDoS, HTTP flood, and ordinary traffic, is utilized to validate the approach proposed by the authors [1]. The taxonomy of various attack types is done using a machine learning technology called WEKA. The attacks from the dataset were classified using J48, MLP, Random Forest, and Nave Bayes, four distinct classifiers. The J48 classifier outperformed the other two classifiers, according on an analysis of the data produced by three different classifiers. J48 provided accuracy of 98.64% whereas the remaining three algorithms, MLP, Random Forest, and Naive Bayes, provided accuracy of 98.63%, 98.10%, and 96.93%, respectively [1]. Finding harmful cyber-attacks requires a monitoring strategy

[2]. Using the statistical indicator continuous ranked probability score (CRPS), as well as the exponential smoothing (ES) method, this study suggests an efficient detection solution for DOS and DDOS assaults. The CRPS is utilized to calculate how different a new finding is from the typical traffic distribution.

This [3] method introduces an innovative hybrid framework built using the data stream method for incrementally identifying DDoS attacks. The proxy side used the multilayer perceptron (MLP), Naive Bayes, decision tree (DTs), random forest, and k-nearest neighbors (KNN) algorithms to enhance results. Over time, using many categorization algorithms as opposed to just one increases your ability to defend against an unanticipated attack.

It is challenging to classify network traffic in high-speed networks in order to recognise DDoS attacks. Intrusion detection systems must use distributed feature selection in parallel computing. The ISCX-IDS, MIT-DARPA, TU-DDoS, and CAIDA datasets were used by the authors to assess their methodology. Our feature ranking algorithm finds the best prospective features from the aforementioned datasets on extensive datasets (50,000–1,000,000 instances) and achieves great accuracy (92–97%) in a parallel setting that requires a lot less time [4].

By enabling the programmability of network components, SDN (Software-Defined Networking) redefines the concept of a network. Network engineers can swiftly monitor, manage, and identify malicious traffic and connection failures on their networks using SDN from a single location. Despite its adaptability, SDN is vulnerable to attacks like DDoS that could paralyse the entire network. The theory proposes using machine learning to distinguish DDoS attack traffic from benign traffic in order to lessen the attack. According to the outcomes, the SVC-RF model, which combines Random Forest and Support Vector Classifier which classifies traffic, achieves the greatest 98.8% testing accuracy and a relatively low rate of false alarms, according to the results [5].

The objective of this project is to simultaneously address imbalance and correlation issues by utilizing a diverse group of multi-label learners. The suggested ensemble technique (EML) is applied to three multi-label data sets that are openly accessible from distinct areas using 18 alternative multi-label classification metrics (Scene, Yeast, and Enron) [6].

The reliability of the Internet is seriously threatened by DDoS attacks, which attempt to overwhelm a target server that has a massive amount of unnecessary traffic from a variety of dispersed and coordinated attack sources. A continuing DDoS attack is therefore challenging to spot. It is crucial to quickly identify changes in resource consumption. Such anomalous alterations might be found statistically. The difficulty with detection using statistics is that it is impossible to accurately predict the distribution of normal network packets. In contrast to statistics-based approaches, clustering methods have the advantage of not requiring any prior knowledge of data distribution. The covariance analysis model makes use of each flag in the control field of the TCP header as a feature. The simulation's findings revealed that this tactic accurately detects SYN flooding attacks in DDoS. Several outcomes of multivariate correlation analysis for detecting DDoS attacks are shown by the tests' high detection accuracy and real-time effectiveness analysis [7].

Attack detection depending on the traffic patterns and identification of attack based on anomalous traffic in traditional network architecture are the two primary DDoS attack

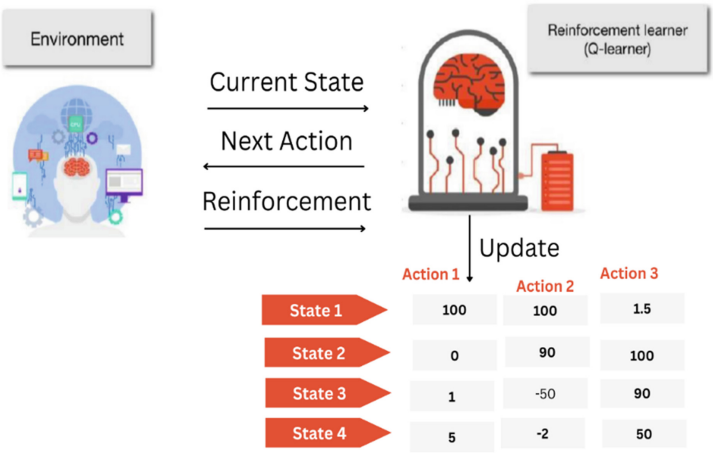

Fig. 4. Architecture Diagram

detection techniques. The most popular implementation strategies include expert systems, state transition, model reasoning, and characteristics matching. The authors used the support vector machine (SVM) algorithm to examine the data and identify DDoS attacks after gathering the six-tuple characteristic values linked to DDoS attacks [8].

4 Methodology

This study's primary goal is to create an automated system that will detect DOS type of cyber-attack happening on end device with the help of reinforcement learning.

The proposed architecture is shown in Fig. 4, where a reinforcement learner (Q-learner) is used to educate the environment to conduct actions in the future through self-directed learning utilizing state-action values or the Q-tab.

To identify and reduce DDoS attacks, a reinforcement learning-based system continuously observers and analyses a variety of variables relating to the load on the server, the dynamic client behaviour, and load on the victim's network. This technique makes use of a unique multi-objective reward function that maximises true positive rates to prevent server crashes when victim system loads are high and minimises false positive rates to avoid collateral damage when victim system loads are low. This method outputs the appropriate course of action for each application message under various conditions. The message may be stopped upstream, stopped locally, or its processing may be delayed. Additionally, this approach gets feedback from the activities taken, enabling it to optimise what actions are ideal in a particular circumstance. For DDoS attack using reinforcement learning, we can consider a set of commands (e.g., ping) as an agent and the environment as a specific network. To set the state we can analyse the incoming flow of packets. The action can be considered by the state in possible range, traffic range, or possible DDoS. If the number of incoming packets exceeds a certain threshold, we set a penalty and if it is in a possible range, we reward it. Software requirements are Wireshark, python libraries, kali Linux.

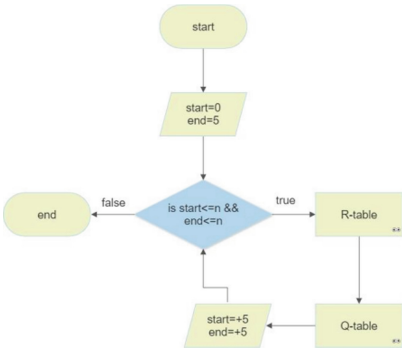

Fig. 5. Base Algorithm

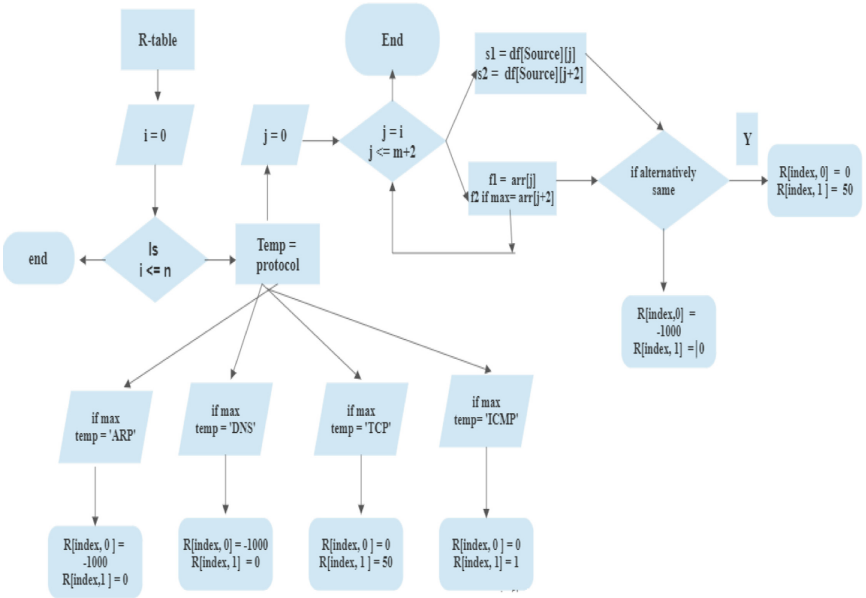

Fig. 6. R-table

4.1 Working Model

See Figs. 5, 6 and 7.

4.2 Implementation

To simulate a DDOS attack, we have used Oracle Virtual Box and Kali Linux. One virtual box represents the attacker, while the other represents the victim. After executing SYN flood attack generation steps using Metasploit, we have captured traffic packets in a PaCap file (Figs. 8 and 9).

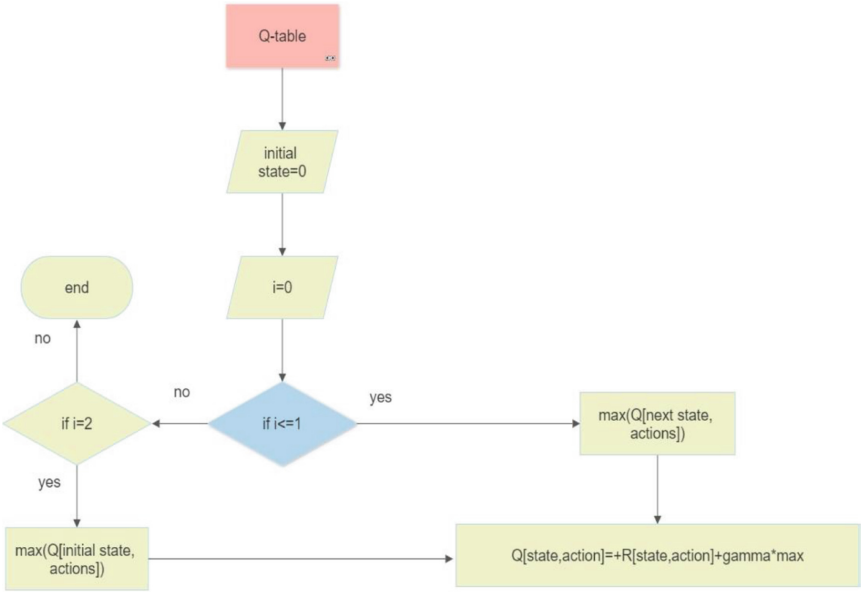

Fig. 7. Q-table

Algorithm 1 Simulation of attack

1: create two virtual machines having kali Linux operating system. One will be target and other the attacker.
2: open the wireshark tool in target machine.
3: go to the attacker machine and enter the following commands:
   a. Get into root privilege
   b. Enter msfconsole
   c. use auxiliary/doc/tcp/synflood
   d. set RHOSTS “ip address”
   e. set RPORT “port no. “
   f. set NUM “no. of syn packets to send”
4: return to the target machine where wireshark is running and start capturing packets.
5: save the pcap and csv files to use later.

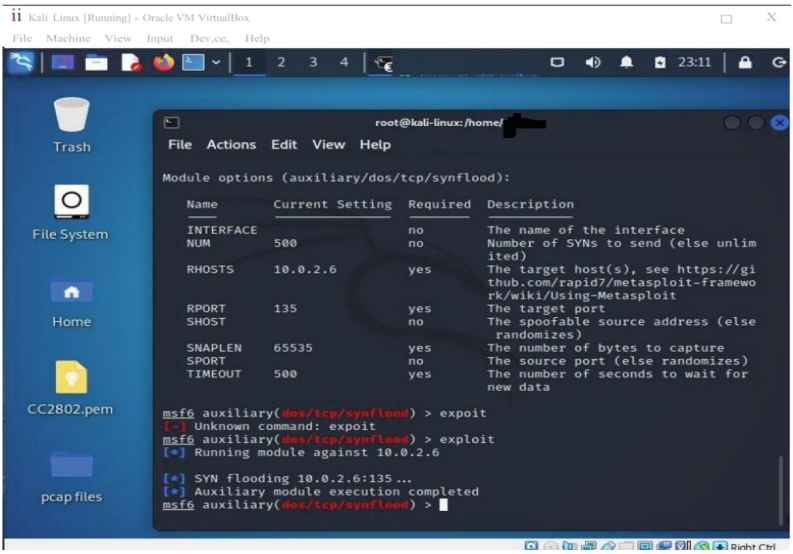

Fig. 8. SYN-flood exploit

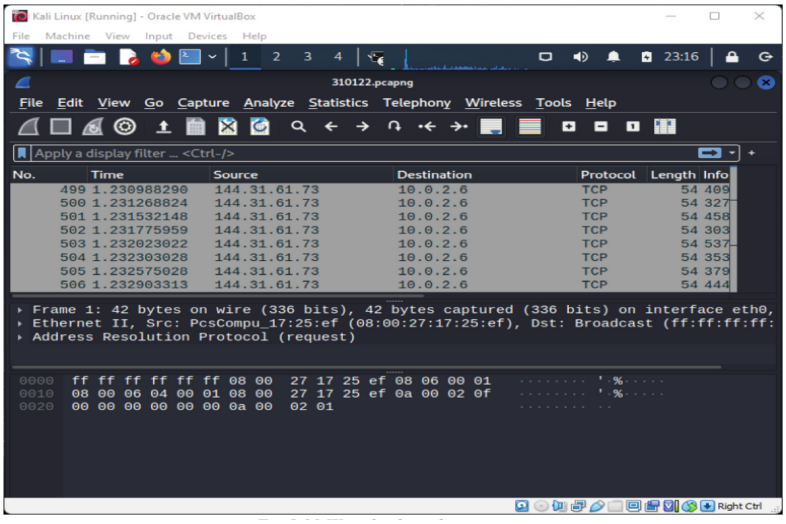

Fig. 9. Wireshark packet capture

Algorithm 2: Detection Process

INPUT: protocol, src/dest, flag

1: input States: Protocol, src/dest, Flag
2: output Actions: Ignore(I), Action(A)
3: policy: Used while R matrix calculation
if protocol is ARP/DNS then
not attack, negative reward, action =ignore (I)
if protocol is TCP/UDP/ICMP then
attack, positive reward, action= Action (A)
if src/destination alternatively same values then
attack, positive reward, action =Action (A)
if src/destination alternatively different values then
not attack, negative reward, action =ignore(I)
if Flag is alternatively same value (flag 2S) then
attack, positive reward, action=Action(A)
else alternatively different values then
not attack, negative reward, action= ignore (I)
4: get R matrix values
5: use R value to calculate Q table
6: at any moment Q table gives the information if the attack has occurred or not.

OUTPUT: Attack or not attack (0 or 1)

4.3 Observation

We have considered time, src, dest, protocol, length, and info columns from the pcap file. Take five rows and compute values for state variables: p-protocol, S/D - source and destination), L-length, and I-info. A server-less attack can be handled in two ways: ignore (I) or alert (A). Reward policy for protocol, ARP = 100 (I), ICMP = -1 (A), DNS = 50 (I), TLS = 25 (I), TCP/UDP = -10 (A). There are three options for S/D and length: different values = +1 (ignore), alternate values = -5 (alert), and same values = -5 (alert). Consider the port number, window size, and presence of the PDU in the pcap file's information column. Here, if port no. is repeated, the reward value is -5 (A), and if port no. is not repeated, the reward value is set at + 2 (I).

According to the action-reward policy, enter values in the R table. For protocols like ARP, DNS, and ICMP, which are rarely used to attack the network, whenever these protocols are encountered, we have given them a negative reward, which indicates no attack. But generally, TCP protocol is used to attack the network, and in our simulation we have also simulated a TCP SYN flood attack, so we have given the maximum positive reward whenever TCP protocol is encountered.

Table 1. Movement of agent /Q-table

Movement of agent	State action reward sum
0	-2100
5	-4200
10	-4945
15	-5160
20	-3995
25	780
30	15005
35	53970
40	157705
45	431010
50	1148255
55	3027750
60	7950055

Apply the Q formula. Enter values in the Q-table. At any instance, we can check the Q-table and predict if there is any kind of attack. Initially, values in the Q table were negative because there was no attack, but after a few packets, values began to increase, indicating that the model had detected the attack. If I(ignore) values are larger positives then no attack. If A (Alert) values are smaller negative then attack possible. After all entries are computed, we can give a result of attack or no attack.

5 Result and Analysis

The agent's present location in the environment will be taken into account when deciding what to do next, in accordance with the Q-learning target. The Q-table is computed using the Bellman Equation, which is employed to choose an agent's optimal course of action. All state and reward values will start out in the Q-table at zero. The data in the Q-table is updated when the agent starts performing an activity. When a correct action is taken, the agent will be rewarded with positive points, and when a mistake is made, the agent will be penalized with negative points, both of which are determined using the Bellman Equation.

Here, Table 1 shows the information regarding agent movement and the state action reward amount that was given to the agent depending on the action taken. The graph in Fig. 10 shows the agent's movement (on the X-axis) versus the state-action reward (on Y-axis). Based on the activities taken by the agent, the graph illustrates how the curve grows from negative values towards positive values.

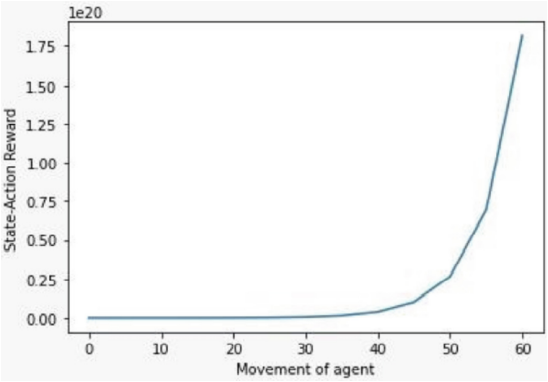

Fig. 10. Plot of Movement of agent vs. State-action reward

6 Conclusion

It is essential to know how dangerous all of our applications may be if left unprotected. Through the use of the lab's resources, this study focuses on various cybersecurity topics. The learning model utilized in the project is not yet an expert, but with further study, it has the potential to become a truly priceless asset for businesses. The goal is to create the ideal network analyzer that will function as a system agent. This agent will have access to the network traffic of a company to define states and reward guidelines. The agent can be trained previously using a specified system so that it can recognize an assault as soon as it happens.

In this study, we simulated an attack and developed a detection strategy that differed from the established methods, which employed supervised and unsupervised machine learning techniques. The agent was given the chance to learn more things through reinforcement learning than they could have through conventional approaches. In this way, the study illustrates how cybersecurity is essential for maintaining a secure environment in the era of growing cloud technologies and computing applications. The beginning of something will eventually contribute to something even better.

References

1. P. S. Saini, S. Behal and S. Bhatia.: Detection of DDoS Attacks using Machine Learning Algorithms. 7th International Conference on Computing for Sustainable Global Development, 2020, pp. 16-21.
2. Benamar Bouyeddou, Benamar Kadri, Fouzi Harrou, Ying Sun.: DDOS-attacks detection using an efficient measurement-based statistical mechanism. Engineering Science and Technology, an International Journal, Volume 23, Issue 4, 2020, Pages 870-878.
3. Soodeh Hosseini, Mehrdad Azizi.: The hybrid technique for DDoS detection with supervised learning algorithms. Computer Networks, Volume 158, 2019, Pages 35-45.
4. Rup Kumar Deka, Dhruba Kumar Bhattacharyya, Jugal Kumar Kalita.: Active learning to detect DDoS attack using ranked features. Computer Communications, Volume 145, 2019, Pages 203-222.

5. Nisha Ahuja, Gaurav Singal, Debajyoti Mukhopadhyay, Neeraj Kumar.: Automated DDOS attack detection in software defined networking. Journal of Network and Computer Applications, Volume 187, 2021.
6. Biggio, B., Fumera, G., Roli, F.: Multiple Classifier Systems under Attack. In: El Gayar, N., Kittler, J., Roli, F. (eds) Multiple Classifier Systems. MCS 2010. Lecture Notes in Computer Science, vol 5997. Springer, Berlin, Heidelberg.
7. Shuyuan Jin and D. S. Yeung.: A covariance analysis model for DDoS attack detection. 2004 IEEE International Conference on Communications, 2004, pp. 1882-1886 Vol.4.
8. Jin Ye, Xiangyang Cheng, Jian Zhu, Luting Feng, and Ling Song.: A DDoS Attack Detection Method Based on SVM in Software Defined Network. 2018 Security and Communication Networks, Hindawi, Volume 2018 |Article ID 9804061.
9. Mallikarjunan, Narasimha & Muthupriya, K. & Shalinie, S. (2016).: A survey of distributed denial of service attack. 1-6. https://doi.org/10.1109/ISCO.2016.7727096. 2016 10th International Conference on Intelligent Systems and Control (ISCO)
10. Sharma, Pratima. (2015).: DDoS Tools: Classification, Analysis and Comparison. June 2015 2nd International Conference on Computing for Sustainable Global Development.
11. Buczak, Anna L. and Erhan Guven.: A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. IEEE Communications Surveys & Tutorials 18 (2016): 1153-1176.
12. Li, J., Qu, Y., Chao, F., Shum, H.P.H., Ho, E.S.L., Yang, L. (2019).: Machine Learning Algorithms for Network Intrusion Detection. In: Sikos, L. (eds) AI in Cybersecurity. Intelligent Systems Reference Library, vol 151. Springer, Cham. https://doi.org/10.1007/978-3-319-98842-9_6