Int. j. inf. tecnol. (February 2024) 16(2):891–900
https://doi.org/10.1007/s41870-023-01509-y

ORIGINAL RESEARCH

FQBDDA: fuzzy Q-learning based DDoS attack detection algorithm for cloud computing environment

Animesh Kumar[1] · **Sandip Dutta**[1] · **Prashant Pranav**[1]

Received: 1 April 2023 / Accepted: 4 September 2023 / Published online: 18 October 2023

Abstract With on-demand resources, flexibility, scalability, dynamic nature, and cheaper maintenance costs, cloud computing technology has revolutionized the Information Technology sector, and almost everyone using the internet relies in some manner on the use of cloud services. Distributed denial of service (DDoS) attack blocks the services by flooding high or low volumes of malicious traffic to exhaust the servers, resources, etc. of the Cloud environment. In today's era, they are challenging to detect because of low-rate traffic and its hidden approach in the cloud. Studying all DDoS attacks with their possible solution is essential to protect the cloud computing environment. In this paper, we have proposed a fuzzy Q learning algorithm and Chebyshev's Inequality principle to counter the problem of DDoS attacks. The proposed framework follows the inclusion of Chebyshev's inequality for workload prediction in the cloud in the analysis phase and fuzzy Q-learning in the planning phase. Experimental results prove that our proposed fuzzy Q-learning based DDoS attack detection algorithm for cloud computing environment (FQBDDA) model prevent DDoS attack.

Keywords Q-learning · Optimization · Fuzzy Q-learning · Fuzzy logic · DDoS attack · Cloud computing · Security

1 Introduction

Cloud computing provides a readily accessible pool of resources, encompassing servers, networks, and user applications, available round the clock. The cloud service provider adopts a "Pay-as-You-Go" approach for all services, allowing consumers to align their choices with budgetary considerations. This approach reduces the time and complexity of procuring diverse software, setting up certificates, and administering software licenses within an organization's local data center. The intricacies of database management, server placement, and data handling are simplified, freeing users from such concerns. This capability empowers small businesses to achieve global expansion economically and efficiently. The services offered by the cloud service provider include platform as a service (PaaS), software as a service (SaaS), and Infrastructure as a Service (IaaS). Some of the advantages of cloud computing are:

Centralized security: It enables the cloud service provider to monitor all network analysis and increase web filtering, traffic analysis, and traffic monitoring. *Cost savings:* It lowers the price of specialized gear. The expense of administration is also decreased. It enables the data center to be run by cloud service providers over time. *Reduced administration:* When we govern the cloud Data, it all happens in one location and is fully managed on the client's behalf. *Reliability:* Cloud users can access their data securely anytime and from anywhere. After cloud technology was introduced, reliance on the other platform was lessened.

A preprint has previously been published [Animesh Kumar et al. 2022] https://assets.researchsquare.com/files/rs-1536879/v1_covered.pdf?c=1663052185.

✉ Prashant Pranav
prashantpranav19@gmail.com

Animesh Kumar
animeshkumarcse@gmail.com

Sandip Dutta
sandipdutta@bitmesra.ac.in

[1] Department of Computer Science and Engineering, Birla Institute of Technology, Mesra, Ranchi 835215, India

Cloud attacks refer to various malicious activities aimed at exploiting vulnerabilities within cloud computing infrastructures for example Man-in-the-middle (MitM) attacks intercept and manipulate data exchanged between users and cloud servers, often leading to unauthorized access and data compromise. Insider threats posed by privileged access individuals also contribute to the risk landscape. Preventing these attacks requires a multi-layered approach of security measures, constant monitoring, and proactive mitigation strategies.

In distributed denial of service attack (DDoS), many machines target users by sending packets with much extra data overhead. Such attacks prevent users from accessing resources by clogging the network with excessive traffic. Following are some DDoS classifications.

HTTP flood attacks: A botnet sends many hyper text transfer protocol (HTTP) requests to overwhelm the web server's resources, including CPU and Memory. In this sort of attack, IP addresses are also a target. It has many categories viz in a *Reflector Attacks* the main goal of this attack is to use third-party reflectors to conceal the attacker's identity. Attackers utilize zombie machines to send traffic to the target system via outside parties using the IP address of the victim's computer. Attacks on the *Domain Name System's Application Layer* try to gain entry through the domain name system's security flaws to amplify small messages into more significant messages. It uses the DNS Flooder tool. The attacker targets the IP address and port address. In *SYN flooding attacks,* attackers use the TCP three-way handshake mechanism to connect to the server in SYN flooding assaults. The target server receives many SYN packets but cannot obtain the last acknowledgment (ACK) needed for the TCP protocol's successful three-way handshake. In *attacks involving user datagram protocol (UDP) flooding* the attacker uses the user datagram protocol to target the server's random ports using IP packets, including UDP packets. In dynamic host configuration protocol (*DHCP) flooding attacks,* the host must issue multiple internet control message protocol (ICMP) destination unreachable packets because the victim system received many UDP packets. Attacks using the ICMP that flood servers with echo requests make them unavailable for authorized users.

Attackers employ a distributed denial of service (DDoS) attack to bar legitimate users from using the services. By sending the victim server numerous queries, the attackers in this attack heavily strain the system. The attacker's massive number of queries completely consumes the victim server's bandwidth, rendering it inaccessible to authorized users. It uses a botnet to use brute force on the network's devices, infecting them with malware. Based on target and behavior, DDoS attacks can be divided into three primary groups. These include application, traffic, and bandwidth attacks.

Attackers use traffic-based attacks to transmit massive amounts of transmission control protocol (TCP) or UDP packets to the victim server, which lowers the victim server's overall performance. In a bandwidth attack, the attackers send significant anonymous data and increase network traffic. An application attack aims to exploit vulnerabilities in the target's application logic, protocols, or resource consumption mechanisms, causing severe performance degradation or even a complete service outage. By overwhelming the application with seemingly legitimate requests, the attacker aims to exhaust its resources, rendering it inaccessible to legitimate users. These attacks often require sophisticated techniques and can be challenging to mitigate, as they can mimic normal user behavior, making it difficult to distinguish malicious traffic from legitimate traffic.

Contribution of the work

- The proposed FQBDDA approach can detect and prevent DDoS attacks on all networking devices associated with the cloud server.
- The proposed model simultaneously deals with the cloud's high-rate and low-rate DDoS attacks.
- This paper also discusses existing techniques to counter DDoS attacks and their impact on cloud services.
- Comparative analysis with other proposed methods is also shown in the paper.

The remainder of the text is organized as follows. The substantial research on DDoS attacks is included in Sect. 2. The suggested work is presented in Sect. 3. The results and discussion section are presented in Sect. 4. The conclusion and future work are included in Sect. 5.

2 Related work

Ray et al. [1] proposed algorithms to mitigate DDoS attacks in mobile healthcare-sensitive data using a cloud-based simulator. The timing factor is also considered in this research paper. Algorithms are formulated to limit the ability of attackers to maneuver within a system. Hnamte et al. [2] proposed a deep convolution neural networks (DCCN) model to counter DDoS attacks in software-defined networks. The result shows an accuracy rate of 99.99% with only a 0.0016 loss rate. InSDN dataset is designed.CIC-IDS2017 and CIC-DDoS2019 datasets are used for training the proposed model. Arunkumar et al. [3] proposed a Gannet optimization algorithm based on machine learning (GOA-optimized hybrid SVM-ELM) in the cloud domain. DoS/DDoS attacks, web-based attacks, port scans, Brute force, Botnet ARES, and normal attacks are classified using this proposed method. The CICIDS2017 dataset is used in experimentation work. Algorithms also counter the problem of intrusion detection in the

cloud. Tripathi et al. [4] proposed a particle swarm optimization (PSO) method for vehicular communication networks (VCO) to mitigate DDoS attacks. The result proves the reduction in transmission delay. Network simulator NS-2 is used for experimental work. The proposed methodology improves the quality of service (QoS) parameter. Najar et al. [5] proposed a machine learning approach using multi-layer perceptron (MLP) and random forest to counter DDoS attacks. The result shows that the random forest algorithm performs a 97% accuracy rate on full test data. MLP performs 74% on the full test dataset. Tinubu et al. [6] developed a decision tree C4.5 algorithm named DT-Model to mitigate DDoS attacks. CICDDoS2019 and FIFA World Cup 1998 datasets were considered for experimental work. The result shows an accuracy rate of 99.7%. Kalnoor et al. [7] proposed a model based on the hidden Markov model (HMM) to counter DDoS attacks in the IoT Domain. The result shows an accuracy rate of 97%. The proposed method, known as variational dynamic Bayesian algorithms, predicts DDoS attacks. Sharma et al. [8] implemented the honeypot and data mining method for mitigating DDoS attacks. Dot net framework is used in this work. An empirical comparison of this method compared to other existing methods is also shown. Future work suggests there is scope in improvement in proposed algorithms.

Mittal et al. [9] proposed a deep learning (DL) method to mitigate DDoS attacks in the cloud. Nearly 34 excellent research publications are considered in this paper. Future research highlights the need for lightweight DL models, the absence of DL validations in real-time DDoS assault scenarios, dynamically updated DL models, and appropriate datasets for accurately detecting DDoS attacks. By lowering the false alarm rate, Arunkumar et al. [10] suggested an intrusion detection technique to address malicious attacks in a cloud computing environment utilizing the machine learning (ML) technique. The outcome demonstrates a decrease in false alarms.

A flow correlation coefficient (FCC)-based protocol-free detection (PFD) approach was proposed by Xiao et al. [11] to identify and stop reflectors' RDDoS attacks. By utilizing this algorithm, attacking flows are recognized. The PFD algorithm protects the application-oriented cloud environment. Future research indicates that FCC flows have room for improvement. A greedy Q-learning technique for reliable resource allocation against a DDOS assault was put forth by Liu et al. [12]. His research primarily focuses on sensor edge cloud networks. probability distribution about the use of the edge VM for resource allocation. Based on Software-defined network (SDN) settings, Aljuhani et al. [13] discussed DDoS mitigation options based on ML/DL techniques. Additionally, it offers ML defenses against DDoS assaults in IoT contexts. Nassif et al. [14] systematically study machine learning methods for cloud security. In this study, KDD and KDD CUP'99 are mainly used. In this study, eleven cloud security domains are identified. SVM employs 30 ML techniques in standalone and hybrid models. In the paper, recent intrusion detection datasets, including CICIDS2017, CSE-CIC-IDS2018, and Kyoto 2006+, are absent. A dynamic DDoS attack detection system based on a classification algorithm, distributed system, and fuzzy logic system was proposed by Alsirhani et al. [15]. The appropriate categorization algorithm can be accurately chosen using the fuzzy logic system. The outcome demonstrates that the suggested algorithm performs well and offers networking device security. software-defined internet of things (Sd-IoT) framework proposed by Yin et al. [16], the cosine similarity of vectors is used to counter DDoS attacks. The DDoS assault is identified using the switch ports. The cyber-physical-social-system (CPSS LR-DDoS) Scenario was proposed by Liu et al. [17]. The suggested algorithm can identify unusual traffic in networking devices. In this paper, surface learning neural networks, K-means, and support vector machines are all utilized. Wang et al. [18] proposed a graphical model to counter DDoS attacks based on the dataset shift problem. The proposed method allows for the detection of real-time network traffic. The network architecture is also shown. Zekri et al. [19] presented the C 4.5 algorithm. The algorithms use signature detection methods, and this study also discusses decision trees. Agrawal et al. [20], a defense strategy that can prevent, detect, and mitigate cloud DDoS attacks, has been presented. In the outcome, performance metrics were also discussed. Phan et al. [21] suggested the enhanced history-based IP filtering scheme (eHIPF) algorithm. It is also utilized in cloud environments built on SDN. The paper's result analysis is effective and novel in defending against DDoS attacks. The dynamic DDoS approach was put up by Li et al. [22] to execute the rules and regulations of container serving for various users. Assis et al. [23] contrasted partial swarm optimization, neural networks, and discrete wavelet transform. In a software-defined network, it is helpful for attack detection and mitigation. A moving target defense (MTD-based) defense plan that uses the GENI Cloud testbed for news and video feeds was suggested by Debroy et al. [24]. The result reveals a 40% decrease in DDoS attacks and a 30% rise in resource usage. Zhijun et al. [25] categorize low-level DDoS attacks and existing defense strategies based on the time and frequency domains of detection and defense. There was a discussion on the inventiveness and aggression of L-DDoS attacks. Aljuhani et al. [13] (2021) developed single and hybrid machine-learning algorithms to counter DDoS attacks in cloud computing.

3 Proposed work

Reinforcement learning's two major pillars are cumulative reward and trial-and-error search. It is based on methods for dynamic programming, which are used to address optimization challenges. Prudencio et al. [26].

Q-learning serves as a link between dynamic programming and reinforcement learning. The Markov Decision problem benefits significantly from the usage of this approach. In the right circumstances, it is utilized to align with optimization values. Instead of using the learning value function, it is based on action-value representation. It has a direct relationship to value iteration. Q-learning uses strategies based on greedy algorithms. The Q-learning agent implements the greedy policy to lower costs while adopting the policy of exploring and exploiting. Ji et al. [27].

Fuzzy Q-learning is an extension of Q-learning in a fuzzy setting. It expands on Watkin's Q-Learning approach for decision-making when the objectives are clear but the system is ill-defined and the problem environment is ambiguous. The state-based state-action pair is the value that this algorithm yields. Therefore, a fuzzy controller must first be created to use fuzzy q learning to detect DDoS attacks. Gheisarnejad et al. [28].

Problem definition: A DDoS attack, one of the most common types of security attacks, hampers a customer's requirement by temporarily blocking the resource requirement of the end user and thus incurs a heavy loss to the cloud service provider. Although many methods have been proposed to mitigate DDoS attack, all depends on training a large dataset. We have proposed a novel method by employing human expert knowledge in a fuzzy controller and training the fuzzy rule base in a Fuzzy Q-learning approach to prevent the occurrence of DDoS attacks in cloud computing.

In this work, we have designed a fuzzy controller to train the fuzzy rules, then formulated nine different fuzzy rules based on expert knowledge Finally, we have used Chebyshev's Inequality to predict workload in a cloud environment from the FIFA 1999 dataset.

3.1 Design of fuzzy controller

We established two input parameters and one output parameter for fuzzy function operations. In the suggested scenario, the two input parameters are the present workload and the current prediction. There are three possible outputs, which we classify as excellent, good, and bad. The current prediction, the first input parameter, is chosen by looking at the historical data on any cloud server. A range of current prediction for various time periods has been defined using Chebyshev's inequality, discussed below in Sect. 3.3. The low, medium, and high fuzzy values are used for the two input parameters. The cloud state may be continuously monitored using the three output states.

If the output status is excellent, monitoring the cloud environment for DDoS attacks is unnecessary. If it is in good state, monitoring is not necessary either. Still, if the output is bad, the system administrator must monitor the cloud environment to see if a DDoS attack has occurred so they can take the necessary precautions. The rules created by expert knowledge utilizing the input and output parameters above are displayed below.

3.2 Fuzzy rules

1. If the current prediction is low and the current workload is low, then an excellent state.
2. If the current prediction is low and the current workload is medium, then good state.
3. If the current prediction is low and the current workload is high, then a bad state.
4. If the current prediction is medium and the current workload is low, then good state.
5. If the current prediction is medium and the current workload is medium, then an excellent state.
6. If the current prediction is medium and the current workload is high, then a bad state.
7. If the current prediction is high and the current workload is low, then a bad state.
8. If the current prediction is high and the current workload is medium, then good state.
9. If the current prediction is high and the current workload is high, then an excellent state.

3.3 Chebyshev's inequality

Is a particular kind of probabilistic inequality. Using the random variable's standard deviation, which is determined from its mean, it provides an upper bound. The fundamental result in the field of statistics is proven using its major result. The weak law of big numbers was also demonstrated using it [29]. It was additionally referred to as Markov's inequality in data analysis. They have a tight connection to one another. Inequality caused by Chebyshev typically involves a random number. All random variables can be used. Using the idea of Chebyshev's inequality, the following equation can be used to define the range of workload predictions as shown in Eq. 1.

$$\begin{aligned} &Mean_{Workload} - 6 \times Standard\ Deviation_{Workload} \\ &\quad < Workload\ Prediction < Mean_{Workload} \\ &\quad + 6 \times Standard\ Deviation_{Workload} \end{aligned} \tag{1}$$

We can forecast a range of workloads for various time slots using the equation above and some historical data. In this instance, the data set is taken from [30]. Table 1 below displays the computed range while ignoring the negative portion.

3.4 FQBDDA: fuzzy Q-learning based DDOS detection algorithm

The two factors we utilized to define the status of the cloud environment, the current prediction (CP) and current workload (CW), are inputs for the suggested algorithm. A reward matrix, an empty q-matrix, and the nine developed fuzzy rules are also supplied as inputs to the proposed FQBDDA in addition to these two. The reward matrix delivers the greatest reward if the agent learns the proper state of the cloud environment and a penalty if the agent learns the incorrect state. Calculating the reward value is as follows shown in Eq. 2.

$$\max\left\{\left(\mathrm{CP_i} + \mathrm{CW_i}\right)/2\mathrm{i} \le \mathrm{N_i}\right\}. \quad (2)$$

The suggested algorithm's output categorizes the cloud environment's condition as excellent, good, or bad.

The q- matrix is initially empty, and after forming nine fuzzy rules, it has expanded to a 9 × 9 empty matrix. The weighted average of the fuzzy rules, as illustrated in step 1 of the procedure below, is used to determine the regulated action from the controller. The q-function is then roughly calculated using the most recent q-values and firing level of rules. The agent performs an action and transitions to the following state after approximating the q-value.

The planned activity is designed to yield the highest possible payoff. The maximum q-value that can be attained in the state is considered when calculating the new state's q-value in step 4. If the maximum reward differs in any way, an error signal is also computed. The q-value is updated at every step.

Equation 3 is to calculate the control action from the logic controller. Equation 4 calculates the q value for the state-action pair *Q (s, a)*. Equation 5 calculates the reward value. Equation 6 derives the value of the new state *s′*. Equation 7 calculates the error signal. Equation 8 shows the updated q values at each step:

Input: current prediction (CP), current workload (CW), reward matrix R, fuzzy rules $\mathrm{Rule_i}$, empty Q- Matrix q(i, a),Leaning rate η = 0:1, discount factor γ = 0:9,

Output: selected state of cloud as excellent state, good state, or bad state

Step 1: for i = 1 to $\mathrm{Rule_i}$ **do**

1. Calculate the control action from the logic controller

$$a = \sum_{i=1}^{N} \mu_i(x) * a_i \quad (3)$$

Where N is the number of rules, $\mu_i(x)$ is the firing strength of rule *i* for input signal *x*, and a_i is the consequent function for the fired rule.

2. Approximate the Q function. The q value for the state-action pair *Q(s, a)* is calculated as:

$$Q(s, a) = \sum_{i=1}^{N} (\mu_i(s) * q\,[i, a_i] \quad (4)$$

Take action and go to the following state s(t+1)

3. Calculate reward value:

$$\mathrm{R_i} = \max\{(\mathrm{CP_i} + \mathrm{CW_i})/2\mathrm{i} \le \mathrm{N_i}\} \quad (5)$$

4. Calculate the value of the new state s': Upon taking action and leaving from state*s* to s', the value of the new state s' is calculated as:

$$V(s') = \sum_{i=1}^{N} \mu_i(s') * \max_{k} (q\,[i, a_k]) \quad (6)$$

Where max $(q\,[i, a_k])$ is the maximum of the q values which can be achieved in the state s'.

5. Calculate the error signal:

$$\Delta QFQL(s, a) = r + {}_{\gamma} V(s') - Q(s, a) \quad (7)$$

Where ${}_{\gamma}$ is the discount rate determining the future reward.

6. Update q values at each step:

$$q\,[i, a_i] = q\,[i, a_i] + \eta.\, \Delta Q.\mu_i(s\,(t)) \quad (8)$$

Table 1 Cloud environment workload prediction using Chebyshev's inequality

Hour	Workload prediction (range)
1	0–24.90
2	0–52.80
3	0–56.30
4	0–57.30
5	0–191.50
6	0–34.80
7	0–138.70
8	0–199.20
9	0–63.50
10	0–50.30
11	0–52.80
12	0–40.10
13	0–52.20
14	0–143.00
15	0–35.80
16	0–40.50
17	0–112.50
18	0–105.00
19	0–108.40
20	0–127.30
21	0–76.20
22	0–49.10
23	0–105.30

4 Result and discussion

We use MATLAB 2023(a) experimental setup. In a Windows Operating system with 16 GB RAM and 1 TB HDD Configuration.

The fuzzy Q learning approach was used to simulate the developed fuzzy rules, and a fuzzy controller was created to regulate the behavior of the intelligent agent. After a few iterations, all the rules were optimized.

Rule 1 states that when both the current workload and the current projection are low, the state is considered excellent. After numerous iterations, the Q values are optimized, and the agent goes to the following state 2 after learning about the state-action pair. In Fig. 1, this is displayed.

Rule 2: When the current prediction is low and the current workload is medium, the state is considered good. After numerous cycles, the Q values are optimized, and the agent proceeds to state 4 after learning about the state-action pair. In Fig. 2, this is displayed.

According to Rule 3, we are in a bad state if the current projection is low and the present workload is high. After 90 iterations, the rule is optimized, and the agent advances to state 7, which is seen in Fig. 3.

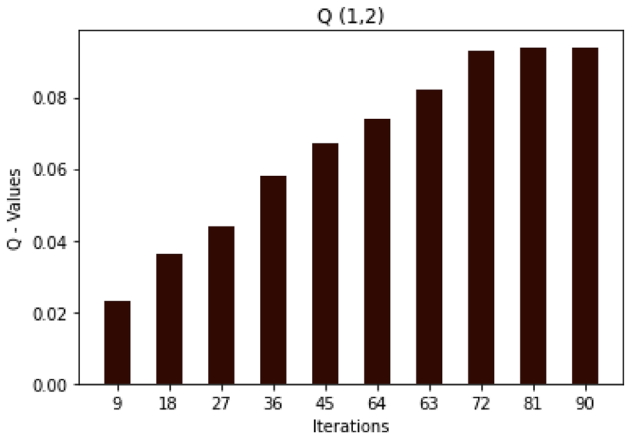

Fig. 1 Optimization of Rule 1

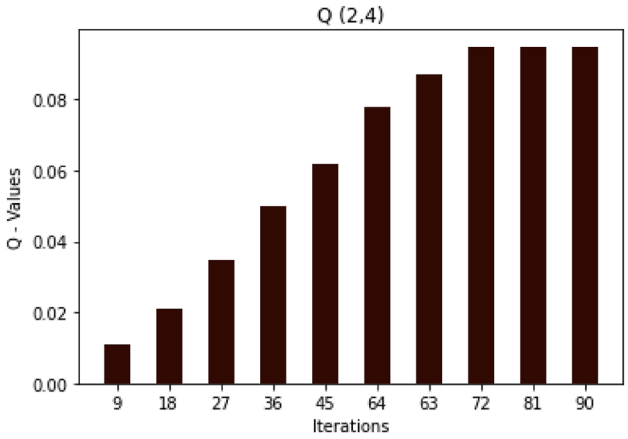

Fig. 2 Optimization of Rule 2

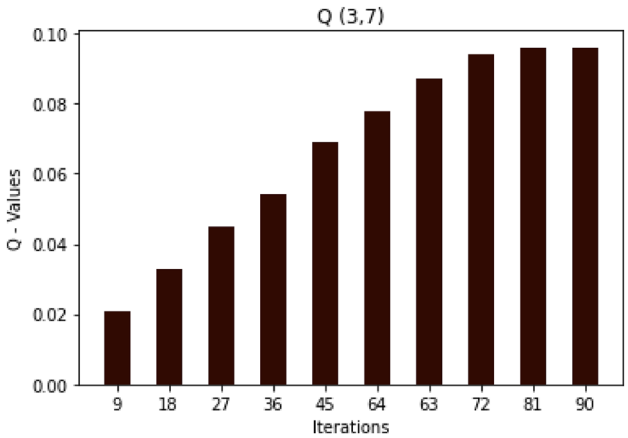

Fig. 3 Optimization of Rule 3

Rule 4: When the present prediction is moderate and the current workload is also low, the state is in good condition. After numerous iterations, the Q values are optimized, and

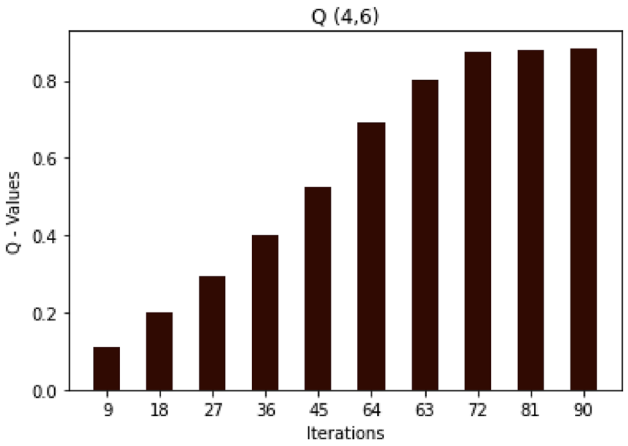

Fig. 4 Optimization of Rule 4

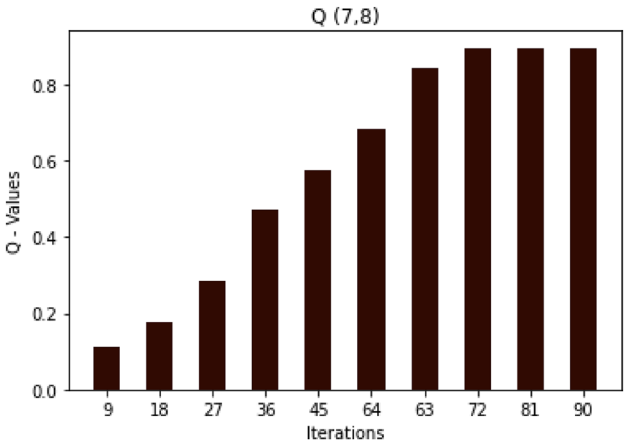

Fig. 7 Optimization of Rule 7

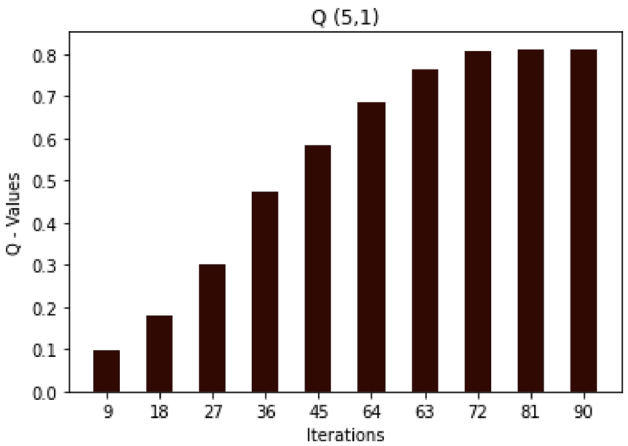

Fig. 5 Optimization of Rule 5

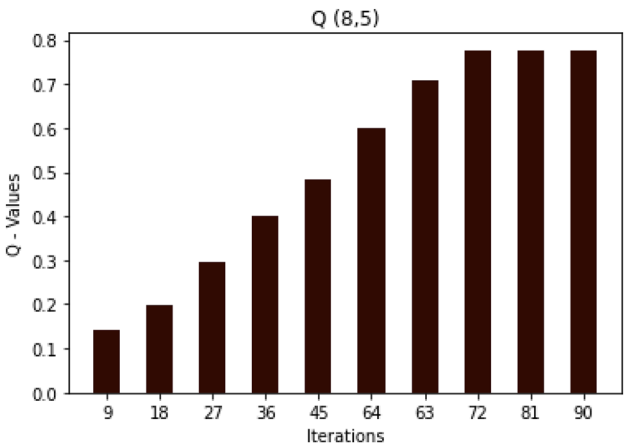

Fig. 8 Optimization of Rule 8

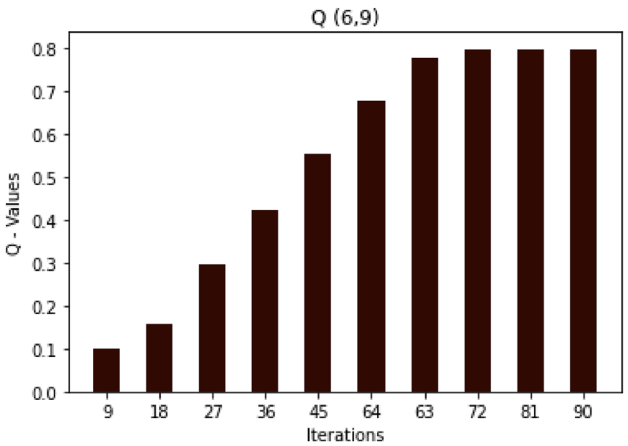

Fig. 6 Optimization of Rule 6

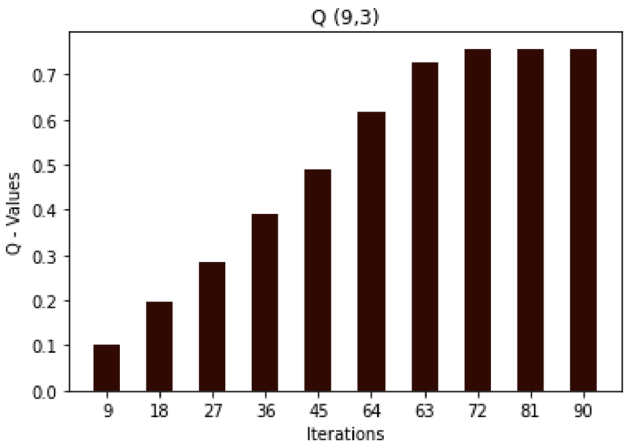

Fig. 9 Optimization of Rule 9

Table 2 Comparison of the work with other existing techniques

Methods	Implementation	Cost	Overhead
Proposed method	Software based	No extra effect on the overall cost	No extra overhead as the method does not require skilled human resources
Hernandez-Suarez et al. [31]	Software based	Overall cost increases due to the use of firewall and CloudShark Firewall	Extra overhead is incurred as the method requires a skilled workforce for CloudShark software and VirusTotal
Sun et al. [32]	Hardware based	Cost increases due to the use of the IEEE 39-bus test system	Extra overhead is incurred as the method requires a skilled workforce
Deng et al. [33]	Software based	Cost increases due to the multi-agent systems	Extra overhead is incurred as the method requires a skilled workforce
Wang et al. [34]	Software based	Cost increases due to the use of sensor nodes	Extra overhead is incurred as the method requires a skilled workforce
Cai et al. [35]	Software based	Cost increases due to the networked control system	Extra overhead is incurred as the method requires a skilled workforce
Xu et al. [36]	Hardware based	Cost increases due to the use of Autonomous ground vehicles	Extra overhead is incurred as the method requires a skilled workforce
Zhang et al. [37]	Software based	Cost increases due to the multi-agent systems	Extra overhead is incurred as the method requires a skilled workforce
Janakiraman et al. [38]	Software based	Cost increases due to using different types of license-based software	Extra overhead is incurred as the method requires a skilled workforce for Owncloud software and VMs

the agent goes to the following state 6 after learning about the state-action pair. In Fig. 4, this is displayed.

Rule 5: When both the present workload and the current prediction are medium, the state is considered excellent. After numerous iterations, the Q values are optimized, and the agent advances to state 1 after learning about the state-action pair. In Fig. 5, this is displayed.

When the present prediction is medium and the current workload is high, rule 6 designates the state as bad. After numerous iterations, the Q values are optimized, and the agent goes to the following state after learning about the state-action pair. In Fig. 6, this is displayed.

When the current prediction is high and the present workload is low, rule 7 declares the situation to be bad. After numerous cycles, the Q values are optimized, and the agent goes to the following state 8 after learning about the state-action pair. In Fig. 7, this is displayed.

Rule 8: When the current prediction is high and the current workload is medium, the state is designated as good. Figure 8 illustrates how the agent learns about the state-action pair and proceeds to the subsequent state when the Q values become more optimal after a number of cycles.

When both the present workload and the current prediction are high, rule 9 declares the state to be excellent. After numerous iterations, the Q values are optimized, and the agent goes to the following state 3 after learning about the state-action pair. In Fig. 9, this is displayed.

Table 2 below shows a comparative analysis of the proposed method with other existing techniques for DDoS attack detection and prevention in a cloud environment.

5 Conclusion

One of the most well-known attacks in a cloud setting is the DDoS attack. In addition to denying the consumer access to the resources, it costs the service provider more money because SLA and QoS are broken. It interferes with the cloud security procedures' purpose of availability. To combat the DDoS attack in the cloud, we suggested a Fuzzy Q-learning-based solution in this study. The agent learns about the ideal cloud environment state through the proposed method and adapts its behavior. The outcome demonstrates that the agent chooses the solution with the highest reward value after a few iterations of learning about the environment.

In the future, other reinforcement learning technique such as SARSA can be used to detect and prevent DDoS attack in a cloud environment. The proposed method using fuzzy Q-learning can be used to detect and prevent other cloud attacks such as SQL injection attack and VM Side channel attack.

Acknowledgements Animesh Kumar has written and prepared the manuscript, which Sandip Dutta and Prashant Pranav have checked. All three authors have formulated the methodology.

Data availability Data related to the ongoing research will be available upon request.

Declarations

Conflict of interest The authors declare that they do not have any conflict of interest.

References

1. Ray S, Mishra KN, Dutta S (2022) Detection and prevention of DDoS attacks on M-healthcare sensitive data: a novel approach. Int J Inf Technol 14:1333–1341. https://doi.org/10.1007/s41870-022-00869-1
2. Hnamte V, Hussain J (2023) An efficient DDoS attack detection mechanism in SDN environment. Int J Inf Technol 15:2623–2636. https://doi.org/10.1007/s41870-023-01332-5
3. Arunkumar M, Kumar KA (2023) GOSVM: gannet optimization-based support vector machine for malicious attack detection in cloud environment. Int J Inf Technol 15:1653–1660. https://doi.org/10.1007/s41870-023-01192-z
4. Tripathi KN, Yadav AM, Sharma SC (2022) DDOS: data dissemination with optimized and secured path using modified particle swarm optimization in vehicular communication network (VCN). Int J Inf Technol 14:1855–1868. https://doi.org/10.1007/s41870-021-00783-y
5. Najar AA, Manohar Naik S (2022) DDoS attack detection using MLP and random forest algorithms. Int J Inf Technol 14:2317–2327. https://doi.org/10.1007/s41870-022-01003-x
6. Tinubu CO, Sodiya AS, Ojesanmi OA et al (2022) DT-Model: a classification model for distributed denial of service attacks and flash events. Int J Inf Technol 14:3077–3087. https://doi.org/10.1007/s41870-022-00946-5
7. Kalnoor G, Gowrishankar S (2022) A model for intrusion detection system using hidden Markov and variational Bayesian model for IoT based wireless sensor network. Int J Inf Technol 14:2021–2033. https://doi.org/10.1007/s41870-021-00748-1
8. Sharma P, Nagpal B (2022) HONEYDOS: a hybrid approach using data mining and honeypot to counter denial of service attack and malicious packets. Int J Inf Technol 14:837–846. https://doi.org/10.1007/s41870-018-0182-4
9. Mittal M, Kumar K, Behal S (2022) Deep learning approaches for detecting DDoS attacks: a systematic review. Soft Comput. https://doi.org/10.1007/s00500-021-06608-1
10. Arunkumar M, Ashok Kumar K (2022) Malicious attack detection approach in cloud computing using machine learning techniques. Soft Comput 26:13097–13107. https://doi.org/10.1007/s00500-021-06679-0
11. Xiao L, Wei W, Yang W et al (2017) A protocol-free detection against cloud oriented reflection DoS attacks. Soft Comput 21:3713–3721. https://doi.org/10.1007/s00500-015-2025-6
12. Liu J, Wang X, Shen S, Yue G, Yu S, Li M (2020) A Bayesian Q-learning game for dependable task offloading against DDoS attacks in sensor edge cloud. IEEE Internet Things J 8(9):7546–7561. https://doi.org/10.1109/JIOT.2020.3038554
13. Aljuhani A (2021) Machine learning approaches for combating distributed denial of service attacks in modern networking environments. IEEE Access 9:42236–42264. https://doi.org/10.1109/ACCESS.2021.3062909
14. Nassif AB, Talib MA, Nasir Q, Albadani H, Dakalbab FM (2021) Machine learning for cloud security: a systematic review. IEEE Access 9:20717–20735. https://doi.org/10.1109/ACCESS.2021.3054129
15. Alsirhani A, Sampalli S, Bodorik P (2019) DDoS detection system: using a set of classification algorithms controlled by fuzzy logic system in apache spark. IEEE Trans Netw Serv Manag 16(3):936–949. https://doi.org/10.1109/TNSM.2019.2929425
16. Yin D, Zhang L, Yang K (2018) A DDoS attack detection and mitigation with software-defined internet of things framework. IEEE Access 6:24694–24705. https://doi.org/10.1109/ACCESS.2018.2831284
17. Liu Z, Yin X, Hu Y (2020) CPSS LR-DDoS detection and defense in edge computing utilizing DCNN Q-learning. IEEE Access 8:42120–42130. https://doi.org/10.1109/ACCESS.2020.2976706
18. Wang B, Zheng Y, Lou W, Hou YT (2015) DDoS attack protection in the era of cloud computing and software-defined networking. Comput Netw 81:308–319. https://doi.org/10.1016/j.comnet.2015.02.026
19. Zekri M, Kafhali SE, Aboutabit N, Saadi Y (2017) DDoS attack detection using machine learning techniques in cloud computing environments. In: 2017 3rd international conference of cloud computing technologies and applications (CloudTech), pp 1–7. https://doi.org/10.1109/CloudTech.2017.8284731
20. Agrawal N, Tapaswi S (2019) Defense mechanisms against DDoS attacks in a cloud computing environment: state-of-the-art and research challenges. IEEE Commun Surv Tutor 21(4):3769–3795. https://doi.org/10.1109/COMST.2019.2934468
21. Phan TV, Park M (2019) Efficient distributed denial-of-service attack defense in SDN-based cloud. IEEE Access 7:18701–18714. https://doi.org/10.1109/ACCESS.2019.2896783
22. Li Z, Jin H, Zou D, Yuan B (2020) Exploring new opportunities to defeat low-rate DDoS attack in container-based cloud environment. IEEE Trans Parallel Distrib Syst 31(3):695–706. https://doi.org/10.1109/TPDS.2019.2942591
23. De Assis MVO, Novaes MP, Zerbini CB, Carvalho LF, Abrãao T, Proença ML (2018) Fast defense system against attacks in software defined networks. IEEE Access 6:69620–69639. https://doi.org/10.1109/ACCESS.2018.2878576
24. Debroy S, Calyam P, Nguyen M, Neupane RL, Mukherjee B, Eeralla AK, Salah K (2020) Frequency-minimal utility-maximal moving target defense against DDoS in SDN-based systems. IEEE Trans Netw Serv Manage 17(2):890–903
25. Zhijun W, Wenjing L, Liang L, Meng Y (2020) Low-rate DoS attacks, detection, defense, and challenges: a survey. IEEE Access 8:43920–43943. https://doi.org/10.1109/ACCESS.2020.2976609
26. Prudencio RF, Maximo MR, Colombini EL (2023) A survey on offline reinforcement learning: taxonomy, review, and open problems. IEEE Trans Neural Netw Learn Syst. https://doi.org/10.1109/TNNLS.2023.3250269
27. Ji Z, Xiao W (2020) Improving decision-making efficiency of image game based on deep Q-learning. Soft Comput 24:8313–8322. https://doi.org/10.1007/s00500-020-04820-z
28. Gheisarnejad M, Sharifzadeh M, Khooban M-H, Al-Haddad K (2023) Adaptive fuzzy Q-learning control design and application to grid-tied nine-level packed E-cell (PEC9) inverter. IEEE Trans Ind Electron 70(1):1071–1076. https://doi.org/10.1109/TIE.2022.3153803
29. Agahi H, Eslami E (2010) A general inequality of Chebyshev type for semi(co)normed fuzzy integrals. Soft Comput 15:771–780. https://doi.org/10.1007/s00500-010-0621-z

30. https://github.com/SWIMProjectUCB/SWIM/blob/master/workloadSuite/FB2009_samples_24_times_1hr_1.tsv. Accessed 14 Dec 2021
31. Hernandez-Suarez A, Sanchez-Perez G, Toscano-Medina LK, Perez-Meana H, Olivares-Mercado J, Portillo-Portillo J, García Villalba LJ (2023) ReinforSec: an automatic generator of synthetic malware samples and denial-of-service attacks through reinforcement learning. Sensors 23(3):1231. https://doi.org/10.3390/s23031231
32. Sun J, Qi G, Chai Y, Zhu Z, Guerrero JM (2023) An adaptive V2G capacity-based frequency regulation scheme with integral reinforcement learning against DoS attacks. IEEE Trans Smart Grid. https://doi.org/10.1109/TSG.2023.3270564
33. Deng C, Meng F, Xie X, Yue D, Che WW, Fan S (2023) Data-driven based distributed fuzzy tracking control for nonlinear MASs under DoS attacks. IEEE Trans Fuzzy Syst. https://doi.org/10.1109/TFUZZ.2023.3289972
34. Wang Y, Wang Z, Zou L, Chen Y, Yue D (2023) Distributed proportional-integral fuzzy state estimation over sensor networks under energy-constrained denial-of-service attacks. IEEE Trans Cybern. https://doi.org/10.1109/TCYB.2023.3288829
35. Cai X, Shi K, Sun Y, Cao J, Wen S, Tian Z (2023) Intelligent event-triggered control supervised by mini-batch machine learning and data compression mechanism for TS fuzzy NCSs under DoS attacks. IEEE Trans Fuzzy Syst
36. Xu Y, Wu ZG, Pan YJ (2023) Perceptual interaction-based path tracking control of autonomous vehicles under DoS attacks: a reinforcement learning approach. IEEE Trans Veh Technol. https://doi.org/10.1109/TVT.2023.3287272
37. Zhang Y, Chadli M, Xiang Z (2023) Prescribed-time formation control for a class of multi-agent systems via fuzzy reinforcement learning. IEEE Trans Fuzzy Syst. https://doi.org/10.1109/TFUZZ.2023.3277480
38. Janakiraman S, Deva Priya M (2023) A deep reinforcement learning-based DDoS attack mitigation scheme for securing big data in fog-assisted cloud environment. Wirel Pers Commun 130(4):2869–2886. https://doi.org/10.1007/s11277-023-10407-2