Type of Article *(Original Article)*

Efficient Resource Allocation in Cloud Computing using Federated Heterogeneous 2Q Learning

Bhargavi K[1], Adarsh M G[2]
[1]*Department of Computer Science and Engineering, Siddaganga Institute of Technology, Karnataka, India.*
[2]*Department of Computer Science and Engineering, Siddaganga Institute of Technology, Karnataka, India.*

Abstract -

Cloud computing is one of the potential data storage platform which offers computing services over the internet. Resource provisioning in cloud is the process of selecting, choosing, deploying, and managing both hardware and software resources by assuring the Quality of Service. Effective utilization of resources without overprovisioning or underprovsioning of resources is a significant challenge. Challenges associated with resource provisioning are complexity in monitoring the usage of resources, inability to develop strong policy enforcement, high automated provisioning cost, poor flexibility, and many more. In this paper a novel federated heterogeneous 2Q learning framework is proposed to draw optimal resource allocation policies. The framework is capable enough of formulating optimal Q value for resource allocation considering the heterogeneous nature of host machines, and virtual machines which exhibit varying kind of resource requirements. . Every agent among set of federated set of agents complement each other in the gathering setup to cover entire heterogeneous local trajectories. The linear speedup of the federated learning agents is capable enough of properly balancing workload across machines and also exhibits high fault tolerance towards attacks. F*rom the experiment results it is observed that the performance of the proposed framework is good with respect to execution time, fault tolerance, energy consumption, and learning rate.*

Keywords – *Resource allocation, federated learning, 2Q learning, cloud computing, Efficiency.*

1. Introduction

Cloud computing platform is one of the potential computing platform which is around us for two decades which provides various benefits in terms of high business efficiency, lower cost of operation, higher capability to handle the large amount of data, and better quality of service. Potential application of cloud include healthcare, industrial manufacturing sector, transportation, civil engineering, and so on [1]. However it is also associated several kind of challenges like poor interoperability, improper allocation of resources, improper governance/control, poor flexibility, worse Quality of Service (QoS) policy, unmonitored demand usage, complexity in the design, vulnerability in operation, and so on [2, 3].

Resource allocation is the strategy of allocating the resource to cloud applications by considering the available cloud infrastructure, Service Level Agreement (SLA), cost of operation, and energy consumption. The cloud service provider is responsible for managing the cloud resources by ensuring the better QoS and higher client satisfaction. The main goal of resource allocation is to maximize the resource utilization ratio, minimize the resource idle ratio, and reduce the energy consumption as much as possible [4, 5]. However the cloud computing architecture is typically composed of variety of heterogeneous host machines, and virtual machines which exhibit varying kind of resource requirements. Because of which resource allocation frameworks are subjected to issues like high energy consumption, poor fault tolerance, improper workload balancing, and so on. Hence there is a need to design a resource allocator by considering all these issues which is a prominent concern of any cloud service providers [6, 7].

Federated heterogeneous Q-learning is one of the prominent form of Q-learning which formulates optimal value for the Q-function by periodically aggregating the local Q-values estimates. The local Q-value estimates are determined by training the Q-learning agent over locally computed Markovian trajectories which are spread across K Q-learning agents. Benefit offered by considering heterogeneity among Q learning agents is that every Q-learning agent gets full coverage of the state space which is usually stringent over a single Q-learning agent setup. Every agent complement each other in the federated gathering setup to cover entire heterogeneous local trajectories. The linear speedup capability of the K learning agents makes its practically suitable for long effective horizon of large state space environment. While formulating the Q values two adjacent Q-values are taken into consideration which mainly reduces the overestimation of the Q values by decomposing the target max operation into target with respect to target selection and target with respect to action evaluation. The accuracy in calculating the action-values improves over time and also results in significant rise in the quality of resource allocation policy formulated over the cloud setup [8, 9, 10].

Several frameworks are available in literature for resource allocation using the technology like deep learning, multiple cognitive agents-based system, reinforcement learning, Q-learning, deep-Q learning, deep Q network, and so on. However they are subjected to several limitations in terms of vanishing gradient problem, poor generalisation capability, lack of transparency, necessity to perform extensive computation, risks of data leakage, difficulty in achieving global optimal solution, unfavourable dependency over the domain parameters, inflexible with large state space environment, and many more [11, 12].

The rest of the paper is organized as follows: Section 2 describes related work carried out in literature. Section 3 gives architecture and algorithm of proposed work. Section 4 deals with experimental setup and results discussion. Finally section 5 draws the conclusion.

2. Related Work

Al-Asaly et al presents a resource allocation model for cloud computing using deep learning [13]. The resource requirements of the client are dynamic in nature which leads to frequent load imbalance. Due to heavy fluctuations in the resource requirement, it is difficult for the cloud provider to forecast the resource requirement in the next time interval. The complexity of workload prediction still gets complicated due to the correlation between the large workloads among the virtual machines. Here an automatic intelligent forecasting mechanism is developed using Diffusion Convolutional Recurrent Neural Network (DCRNN). The main goal is to improve the forecasting accuracy and minimize the error between predicated workload and actual workload. The DCRNN is capable enough of capturing the spatial dependency by performing bidirectional random walk over the graph and temporal dependency using scheduled sampling. However training time taken is too long and it also suffers from vanishing gradient problem.

Ali et al presents a new multi-agent based reinforcement learning model for efficient allocation of resources in the cloud [14]. Meeting the customer requirement is difficult in cloud environment because of heterogeneous nature of the cloud nodes. Here the computing capabilities of multiple agents is combined with the Q-learning agent to enhance the performance of the cloud resource allocation technique. Multiple agents are used to allocate the resources as per the varying customer requirements. After that Q learning policies are formulated using which virtual machine always move best state based on the current state environment. Important parameter considered for optimization purpose is weight of the virtual machine, energy usage, and quantum time. It provides best action for every changing state of the cloud environment. The performance of the proposed model is good in terms of execution time, energy consumption, and load balancing ratio is testbed setup. However the proposed model is not tested over the real cloud computing scenario, which limits its applicability.

Uma et al discuss an optimized approach for resource allocation in cloud using deep Q-learning algorithm [15]. Cloud service providers usually perform virtual machine leasing to reduce the energy consumption of the virtual machines. Even the cloud service providers face challenges related to energy consumption and scalability. Even task dependencies need to be considered for parallel execution of tasks. Here deep Q-learning is performed to approximate Q-function which helps in determining optimal action at any given state. It combines Q-learning with neural network to find optimal Q-value for all possible actions as the output. It is capable enough of handling resource allocation problem using stochastic transition and rewards without any sort of adaption. The experimental results ensure that the performance of the proposed technique in energy consumption, and scalability. However the size of Q table grows exponentially with the varying number of states and actions. Because of which the resource allocation model becomes infeasible overtime and also involve high computational complexity. Poor generalisation happens due to overfitting of the training data.

Chen et al discuss the application of deep Q-learning network for resource allocation in cloud [16]. The main aim of resource allocation technique is to provide optimal amount of resources to the tasks so that it gets completed within lesser time span. Here optimal allocation of tasks onto resources using deeply nested Q-learning function is proposed. At first a resource allocation model is designed and even the goal function is designed. A deep neural network is designed to represent the Q-function instead of maintaining an ordinary table of values. It is capable enough of achieving the trade-off between exploration and exploitation activities. From the results it is observed that it has many advantages in terms of CPU utilization, shortened scheduling time, and strong ability to balance the load across the nodes. However the resource allocation policies lack transparency feature. It is computationally expensive and consume enormous amount of energy.

Ebadi et al proposes a supervised learning strategy to allocate the resources in cloud for effective data transmission [17]. A backtracking regularized model is developed for effective allocation of resources among cloud nodes. A machine learning model is designed to train, test, and classify the resource allocation. Supervised learning model and optimization mechanism provides flexibility in operation. It takes the basis from fundamental principle of natural selection process and genetic variation to iteratively refine the resource allocation policies. The machine learning approach is combined with prediction, backtracking, and optimization which aids in agile resource allocation that even boost the data transmission. First the machine learning model is trained using historical data to predict the future demand of resources. Further regularization is applied to fit the training data well and not allowing the model to choose lot of data complexities. Most widely used to regularization method is L2 regularization or ridge regularization. Experimental results shows that the resource utilization is high and the latency incurred is less.

3. Proposed Work

A high level architecture for resource allocation in cloud computing using federated H2Q learning is shown in Figure 1. The architecture is composed of four layers. First layer is composed of set of physical machines. Second layer is virtual machines composed of set of virtual machine. Every physical machine host set of virtual machines. The proposed H2Q resource allocator sits in third layer. Finally the fourth later is composed of set of users. H2Q resource allocator is responsible for accepting set of incoming task requests from the user. Then properly allocate the resources for each of the tasks. Resource allocation policies are formulated using heterogeneous double state federated Q-learning mechanism. Two main operations carried out by the federated Q learning agents are local updates and the other one is periodic averaging. During local updates, each agent update the Q table entries independently using update rule. During periodic averaging the intermediate estimate of Q values are averaged by the server to determine the final estimate of the Q values. Even the generality is not loosen here as the total number of iterations considered for training of federated H2Q agents is divisible by the number of rounds of communication.

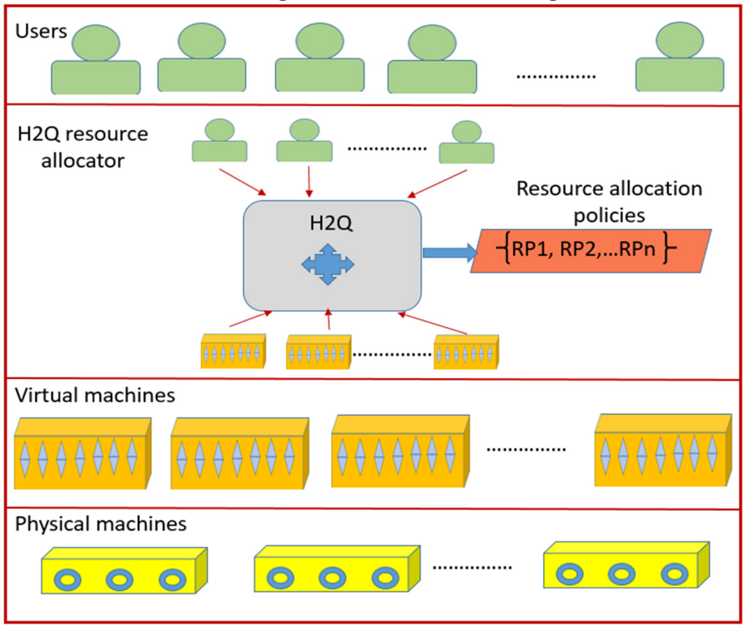


Figure 1: High level architecture of federated H2Q resource allocator

The detailed working of the proposed federated H2Q resource allocator is shown in Algorithm 1. The algorithm alternates between local updating of agents and periodic averaging at the central server. It begins with the initialization of the local Q function $Q_0^K = Q_0$, for all $k \in [K]$. Every agent will maintain an estimate of local Q function and a local Q value function.

$$\forall_{s\in S}: \; V_i^k(s) \coloneqq max_{a\in A} Q_i^k(s,a)$$

The local updates and periodic averaging operations of the federated H2Q learning agents are as follows:
Local updates: Every agent updates all Q values in the Q table to determine the indeterminate estimate of the Q value.

$$\forall(s,a) \in S * A: \; Q_{i-\frac{1}{2}}^k(s,a) = (1-\eta)Q_{i-1}^k(s,a) + \eta\left(r(s,a) + \gamma V_{i-1}^k\left(s_t^k(s,a)\right)\right)$$

Periodic averaging: The intermediate Q values obtained during local updates phase are periodically averaged by the server to formulate updated estimated Q_i^k at the end of the Ith iteration of Q-learning phase.

$$\forall(s,a) \in S * A: Q_i^K(s,a) = \begin{cases} 1/k \sum_{k=1}^{k=K} Q_{i-1/2}^k(s,a), if \; i = 0(mod \; \tau) \\ Q_{i-1/2}^k(s,a), otherwise \end{cases}$$

The maximum number of iterations considering for Q learning is I, the algorithm generate the final Q-value as output which is average over all local Q estimates i.e. $Q_i = 1/k \sum_k Q_I^k$, in which the value of I is divisible by τ and the number of rounds of communication . $C_{round}=\frac{I}{\tau}$.

Algorithm 1: Working of the proposed Federated Heterogeneous 2Q Learning for resource allocation

1. **Start**

2: **Input:** User task requests $UT = \{ut_1, ut_2, \dots, ut_m\}$, virtual machine resources $VM = \{vm_1, vm_2, \dots, vm_n\}$, learning rate η , number of Q learning agents K, synchronization period T, total number of iterations I

Output: Resource allocation Policies

RAP= $\{rap_1, rap_2, ., rap_p\}$.

3: Initialize $Q_0^K = Q_0$, for all $k \in [K]$

3: **For** each iteration $i = 1, i = 2, \dots . i = I$ do

4: Take action $a_i^{k-1} \cong \pi_b^k(s_{i-1}^k, a_{i-1}^k)$, collect the reward $r_i^{k-1} \cong \pi_b^k(s_{i-1}^k, a_{i-1}^k)$, reach the next state $s_i^k \cong P(.\,|s_{i-1}^k, a_{i-1}^k)$

5: Compute $Q_{i-1/2}^K$ according to the equation below

6: $$Q_{i-\frac{1}{2}}^K(s,a) = \begin{cases} (1-\eta)Q_{i-1}^k(s,a) + (1-\eta)Q_{i-1}^k(s,a) + \eta(r_i^{k-1} + \gamma V_{i-1}^k(s_i^k)), \\ \quad if(s,a) = (s_{i-1}^k, a_{i-1}^k) \\ Q_{i-1}^k(s,a), otherwise \end{cases}$$

7: Compute Q_i^K according to the equation below

8: $$Q_i^K(s,a) = \begin{cases} \sum_{k=1}^K \alpha_i^k(s,a)Q_{i-\frac{1}{2}}^K(s,a) + \alpha_i^k(s,a)Q_{i-\frac{1}{2}}^K(s,a),\ if\ i = 0(mod\ \tau) \\ Q_{i-1/2}^k(s,a) + Q_{i-1/2}^k(s,a), otherwise \end{cases}$$

9: **End For**

10: Return:

$Q_i(s,a) = \sum_{k=1}^{k=K} \alpha_i^k(s,a)Q_i^K(s,a), all\ (s,a)\epsilon S * A$

11: Generate resource provisioning policies

RAP= $\{rap_1, rap_2, ., rap_p\}$.

12: Classify the resource provisioning policies into three different categories based on the maximum cumulated reward value.

$$r_\theta = \max(r_i^{k-1}, r_j^{k-1}, \dots, r_k^{k-1})/I$$

13: **FOR** each of the $rap_1 \in$ RAP **do**

14: **IF** $(rap_i < r_\theta/2)$

Less flexible resource provisioning policies

$Less\_rap_1 \Leftarrow Q_i(s,a) = \sum_{k=1}^{k=K} \alpha_i^k(s,a)Q_i^K(s,a),\ \ all\ (s,a)\epsilon S * A$ :: {Unable to handle large state space, highly sensitive towards optimization , parameters, speedup is slow, long effective horizon}

15: **ELSE IF** $(rap_i \cong r_\theta/2)$

Moderate flexible resource provisioning policies

$Mod\_rap_1 \Leftarrow Q_i(s,a) = \sum_{k=1}^{k=K} \alpha_i^k(s,a)Q_i^K(s,a),\ all\ (s,a)\epsilon S * A$ {Unable to handle large state space, acute sensitivity towards optimization parameters, speedup is moderate, moderately poor horizon}

16: **ELSE**

Highly flexible resource provisioning policies

$High\_rap_1 \Longleftarrow Q_i(s,a) = \sum_{k=1}^{k=K} \alpha_i^k(s,a) Q_i^K(s,a), \quad all\ (s,a) \epsilon S * A$ {able to handle large state space, No sensitivity towards optimization parameters, speedup is high, small effective horizon}

17: **END IF**

18: **END FOR**

19: Return

$High_{rap} = \{High\_rap_{1,} High\_rap_{j,}, \ldots High\_rap_k\}$

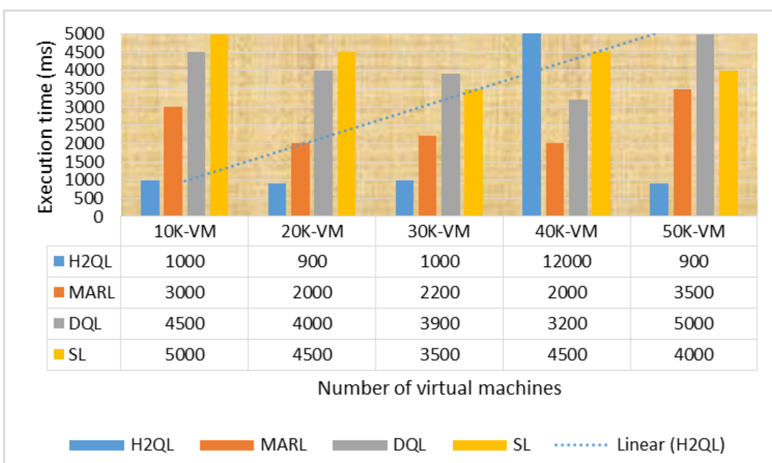
20: **Stop**

---

4. Results and Discussion

For simulation purpose CloudSim 3.3 simulator is used. The simulation parameter setup is as follows: user (Number of users=25, number of brokers=05), host machine (Number of host machine=1000-10000, RAM=3048, storage=10,000, bandwidth=10,000), virtual machine (Number of virtual machine=10000 to 50000, type of policy=Round robin, RAM=2048, Bandwidth=20,000, MIPS=10^5K MIPS, size=100MB, Operating system=Windows/ubuntu, Number of CPUs=20), Task (Task distribution=exponential, task computing requirement=10^6MIPS, task size=8500 Bytes, task output size=25*10^6 MIPS, task duration=1Sec), host machine initial load is 30%, server computing speed=10^6MIPS, power management technique=no, and simulation time=60 seconds, average data processing time for applications in cloud=180-200ms, data sensing interval of IoT sensors=200-600ms. Applications=E-Health, compute-intensive application, smart home, augmented reality. The performance of the proposed federated H2Q learning is compared with the recent existing works multi agent reinforcement learning [14], deep Q learning [15], and supervised learning [17].

Execution time



	10K-VM	20K-VM	30K-VM	40K-VM	50K-VM
H2QL	1000	900	1000	12000	900
MARL	3000	2000	2200	2000	3500
DQL	4500	4000	3900	3200	5000
SL	5000	4500	3500	4500	4000

Figure 2: Number of virtual machines Versus Execution time (ms)

A graph of number of virtual machines versus execution time is shown in Figure 2. It is observed from the graph that the execution time of H2QL is consistently less. As its learning efficiency is high by employing Q-learning variants over various virtual machine configurations. The execution time of MARL is moderate over the increasing number of virtual machines. The execution time of DQL and SL are very high

Energy consumption

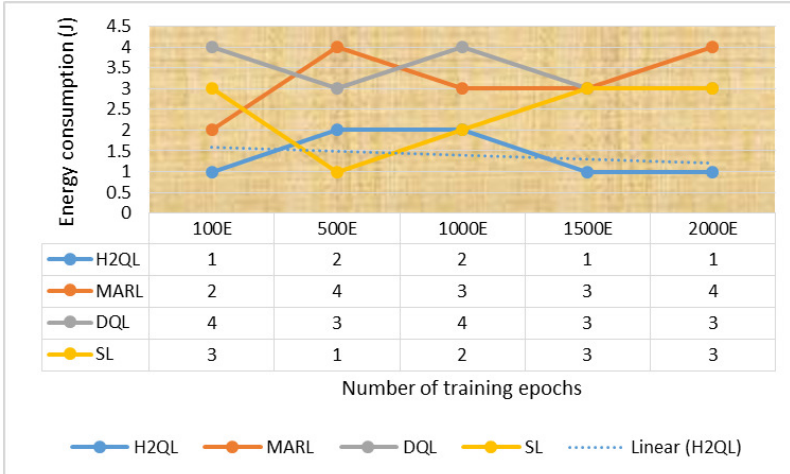


Figure 3: Number of training epochs Versus Energy consumption (J)

A graph of number of epochs of training versus energy consumption is shown in Figure 3. It is known from the graph that the energy consumption of H2QL is less. As it cannot highly adapt complex environment by efficiently handling different aspects of the environment. The energy consumption of MARL and DQL are high. As it require large amount of labelled to attain very high level of performance. The energy consumption of SL is moderate as it suffer from overfitting problem in which the model perform well on training data but fails over unseen data.

Fault tolerance

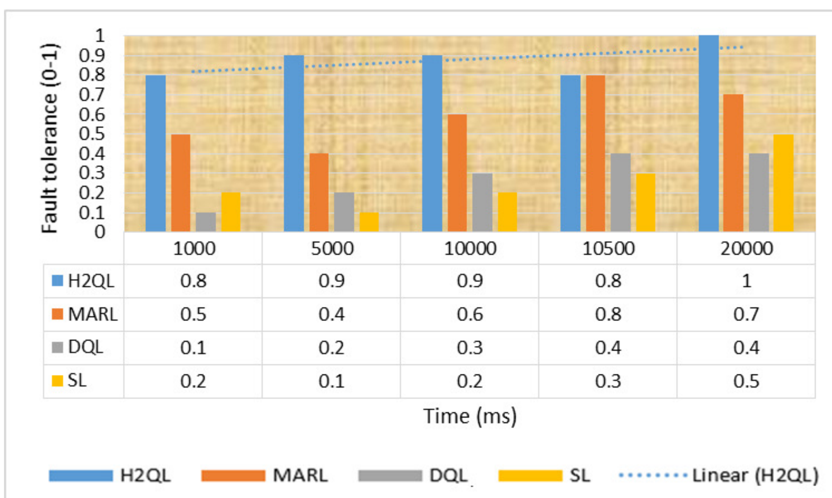


Figure 4: Time versus Fault tolerance

A graph of time versus fault tolerance is shown in Figure 4. The fault tolerance of H2QL is consistently high with respect to time. As it can effectively model the heterogeneous Q-learning setup by optimizing resource provisioning objectives. The fault tolerance of MARL is moderate due to complexity in coordination and demands highly sophisticated coordination strategies. The fault tolerance of DQL and SL are low with respect to time due to limited generalization as it is highly depended on the quality and diversity of the training data.

Learning rate

A graph of number of raining epochs versus learning rate is shown in Figure 5. The learning rate of H2QL is consistently very high. As it can easily integrate among various types of agents enabling a more tailored approach to learning and resource provisioning decision making. The learning rate of MARL is above moderate. As frequent coordination among the agent's results in unpredictable behaviour of the agent which leads to difficulty in ensuring the desired solutions. The learning rate of DQL and SL are low whenever there are chances of bias in the training data which leads to unfair results or discriminatory outcomes.

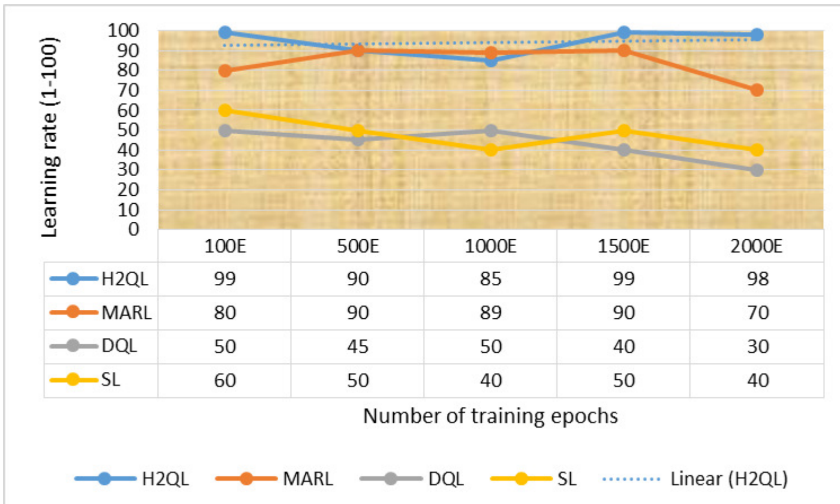


Figure 5: Number of training epochs Versus Learning rate

5. Conclusion

In this paper a novel federated heterogeneous 2Q learning framework is proposed for resource provisioning in cloud environment. The resource provisioning polices are determined by considering the combination of federated learning with double Q states of the learning agents. The overestimation of Q learning policy is prevented by considering the two adjacent Q states before arriving at the decisions. The performance results proves that the framework achieves lower energy consumption, high fault tolerance, high learning rate, and lower execution time. As future work analytical modelling of the proposed framework will be done in finite and infinite cloud setup. Also the thorough investigation of the framework towards automatic scale up and scale down with the varying flux in the demand of resources will be analysed.

Conflicts of Interest There is no possible conflict of interest with respect to the submitted manuscript.

Funding Statement

No financial support is received for the submitted manuscript by any funding bodies.

Acknowledgments

I acknowledge the support extended by Siddaganga Institute of Technology during the research and preparation of the manuscript. Both the authors Bhargavi K and Adarsh M G contributed equally to this work.

References

[1] Abid, A., Manzoor, M. F., Farooq, M. S., Farooq, U., & Hussain, M. (2020). Challenges and Issues of Resource Allocation Techniques in Cloud Computing. *KSII Transactions on Internet & Information Systems*, *14*(7).

[2] Saidi, K., & Bardou, D. (2023). Task scheduling and VM placement to resource allocation in Cloud computing: challenges and opportunities. *Cluster Computing*, *26*(5), 3069-3087.

[3] Mohammad, A., & Abbas, Y. (2024). Key Challenges of Cloud Computing Resource Allocation in Small and Medium Enterprises. *Digital*, *4*(2), 372-388.

[4] Belgacem, A. (2022). Dynamic resource allocation in cloud computing: analysis and taxonomies. *Computing*, *104*(3), 681-710.

[5] Abedi, S., Ghobaei-Arani, M., Khorami, E., & Mojarad, M. (2022). Dynamic resource allocation using improved firefly optimization algorithm in cloud environment. *Applied Artificial Intelligence*, *36*(1), 2055394.

[6] Abedi, S., Ghobaei-Arani, M., Khorami, E., & Mojarad, M. (2022). Dynamic resource allocation using improved firefly optimization algorithm in cloud environment. *Applied Artificial Intelligence*, *36*(1), 2055394.

[7] Petrovska, I., & Kuchuk, H. (2023). Adaptive resource allocation method for data processing and security in cloud environment. *Advanced Information Systems*, *7*(3), 67-73.

[8] Qi, Y., Wang, X., Li, H., & Tian, J. (2024). Leveraging Federated Learning and Edge Computing for Recommendation Systems within Cloud Computing Networks. *arXiv preprint arXiv:2403.03165*.

[9] Messaoud, S., Bradai, A., Ahmed, O. B., Quang, P. T. A., Atri, M., & Hossain, M. S. (2020). Deep federated Q-learning-based network slicing for industrial IoT. *IEEE Transactions on Industrial Informatics*, *17*(8), 5572-5582.

[10] Zarandi, S., & Tabassum, H. (2021, June). Federated double deep Q-learning for joint delay and energy minimization in IoT networks. In *2021 IEEE International Conference on Communications Workshops (ICC Workshops)* (pp. 1-6). IEEE.

[11] Belgacem, A., Mahmoudi, S., & Kihl, M. (2022). Intelligent multi-agent reinforcement learning model for resources allocation in cloud computing. *Journal of King Saud University-Computer and Information Sciences*, *34*(6), 2391-2404.

[12] Chen, X., Zhu, F., Chen, Z., Min, G., Zheng, X., & Rong, C. (2020). Resource allocation for cloud-based software services using prediction-enabled feedback control with reinforcement learning. *IEEE Transactions on Cloud Computing*, *10*(2), 1117-1129.

[13] Al-Asaly, M. S., Bencherif, M. A., Alsanad, A., & Hassan, M. M. (2022). A deep learning-based resource usage prediction model for resource provisioning in an autonomic cloud computing environment. *Neural Computing and Applications*, *34*(13), 10211-10228.

[14] Belgacem, A., Mahmoudi, S., & Kihl, M. (2022). Intelligent multi-agent reinforcement learning model for resources allocation in cloud computing. *Journal of King Saud University-Computer and Information Sciences*, *34*(6), 2391-2404.

[15] Uma, J., Vivekanandan, P., & Shankar, S. (2022). Optimized intellectual resource scheduling using deep reinforcement Q-learning in cloud computing. *Transactions on Emerging Telecommunications Technologies*, *33*(5), e4463.

[16] Chen, Z. (2020). Cloud resource allocation based on deep q-learning network. In Data Science: 6th International Conference of Pioneering Computer Scientists, Engineers and Educators, ICPCSEE 2020, Taiyuan, China, September 18-21, 2020, Proceedings, Part I 6 (pp. 666-675). Springer Singapore.

[17] Ebadi, M. E., Yu, W., Rahmani, K. R., & Hakimi, M. (2024). Resource Allocation in The Cloud Environment with Supervised Machine learning for Effective Data Transmission. *Journal of Computer Science and Technology Studies*, *6*(3), 22-34.