Q-Learning-Driven Framework for Optimized Virtual Machine Placement in Cloud Data Centers

Vipin Kumar Jaiswal[1], Jameel Ahmad[2], Kiran Deep Singh[3]

[1,2]Department of Computer Science and Engineering, Integral University, Lucknow, Uttar Pradesh, India
[3]Chitkara University Institute of Engineering and Technology, Chitkara University, Rajpura, Punjab, India

In modern cloud data centers, efficient virtual machine placement is crucial for optimizing resource utilization, reducing computational costs, and enhancing overall system performance. Traditional approaches, often rooted in heuristic and rule-based strategies, struggle with scalability and adapting to dynamic workloads. An innovative framework for virtual machine deployment is presented in this research, utilizing the Q-learning algorithm, a type of reinforcement learning intended to maximize decision-making under dynamic and unpredictable circumstances. Through the use of Q-learning, our framework minimizes resource contention and distributes loads among physical servers by automatically learning the best placement techniques through interactions with the cloud environment. The Q-learning strategy improves resource allocation efficiency by continuously adapting to changing situations, in contrast to static models. The suggested framework shows notable improvements in lowering makespan about 92% that is highly optimizing the resources utilization and lowering computational costs about 34%, thus framework enhancing overall system performance through comprehensive simulations and comparisons with traditional models. The outcomes demonstrate how Q-learning algorithm can improve virtual machine placement tactics and offer a flexible and scalable approach to cloud infrastructure management.

Keywords: Cloudlet, Virtual Machine, Virtual Machine Placement, Cloud Data Centers, CloudSim.

1. Introduction

With a particular emphasis on the effective placement of virtual machines (VMs) within data centers, the optimized VM Placement tackles the core issues related to resource management in cloud environments. Effective virtual machine deployment is crucial in cloud computing due to the dynamic nature of resource requirements, the wide range of applications, and the large number of workload demands. By arranging VMs strategically, cloud providers can maximize resource utilization, save operating costs, enhance system performance, and contribute to energy conservation. Each of these elements is essential to data center management [1].

Traditional virtual machine deployment solutions usually rely on heuristic or rule-based techniques, which can be wasteful and inflexible in dynamic and expanding cloud systems. These methods could lead to excessive energy use, resource fragmentation, and underutilization. Machine learning-based methods, particularly reinforcement learning, provide a viable alternative since they can gradually learn from and adapt to their environment. Among these, Q-learning is a well-known reinforcement learning algorithm that is notable for its simplicity and effectiveness in decision-making tasks. The VM placement task is represented by the method as a Markov Decision Process (MDP), where the data center's status (workload, power consumption, and resource availability) influences the placement decision [2].

Q-Learning improves cloudlet (task)-to-VM scheduling by utilizing reinforcement learning (RL), which allows the algorithm to learn from interactions with the cloud system and progressively optimize the scheduling policy. Models are not necessary for the Q-Learning reinforcement learning method to function. It is predicated on the notion that an agent engages with its environment in order to learn the optimal course of action via trial and error. The agent is rewarded according to the actions taken, and this is done in order to update a Q-table that forecasts the expected future rewards of state-action pairings [3][4].

The proposed framework consists of three main components:

(a) State Representation: This part keeps track of the current state of the data center, including workload distribution, CPU, memory, and storage availability, along with other factors that influence the placement of virtual machines. Because it provides the Q-learning agent with the contextual data it needs to make decisions, this state representation is crucial.

(b) Action Space: In this paradigm, the action space defines the collection of possible activities that the agent can perform. Every step corresponds to configuring a virtual machine on a certain physical server, accounting for the state representation. The challenge is to efficiently explore the action space to avoid resource conflicts, balance load, and determine the optimal placement method.

(c) Reward Function: The incentive function was developed to assist the agent in accomplishing the objectives of the VM placement task. Among the factors that are commonly included in this function are SLA (Service Level Agreement) compliance, power consumption, load balancing, and resource utilization efficiency. Conflicting objectives, such as maintaining performance while lowering power consumption, can be balanced by modifying the reward mechanism.

Last few years, research on cloud computing trying to make it convenient to maximize the use of resources, task scheduling, cloud security, minimize costs, and enhance the performance of overall cloud [5][6][7]. Well-performing resources can be distributed to data centers via cloud computing-enabled cloud service providers (IAAS). Jobs given to several virtual machines operating in parallel are designed to optimize resource utilization. Cloudlets react in a virtual world. When assigning tasks to virtual machines (VMs), it is necessary to take into account and observe factors like success rate, cost, duration, and makespan. To optimize cloud resources and their scheduling, a lot of study has looked at QoS parameters. Virtual machines are scheduled in parallel in order to maximize processing times through the use of minimal resources.

Researchers have been investigating a variety of strategies, including metaheuristic algorithms and reinforcement learning techniques, in response to these difficulties. These strategies seek to capitalize on the advantages of machine learning methods such as Q-learning, which facilitate adaptive decision-making in dynamic contexts, and optimization algorithms, which are excellent at exploring solution spaces. With the help of the Q-learning algorithm, the framework can adapt to the shifting workload and resource availability, providing a dependable solution for the placement of virtual machines in cloud data centers. The system continuously enhances its placement strategy by learning from past decisions, which raises resource consumption and cost effectiveness. Additionally, by employing a reinforcement learning-driven methodology, the framework can handle the complex, non-linear interactions between virtual machines and physical resources that are often challenging for traditional optimization algorithms to address [8].

To put it briefly, a Q-learning-driven architecture for virtual machine placement offers a scalable, adaptable, and efficient method of managing cloud data centers, enabling providers to improve resource efficiency, satisfy performance criteria, and use less energy. In addition to addressing the shortcomings of traditional VM placement strategies, our approach opens up possibilities for autonomous, astute optimization research.

2. Problem Statement

Effective VM deployment is essential in cloud data centers to guarantee optimal resource usage, lower operating costs, and uphold service-level agreements (SLAs). Assigning virtual machines to physical machines (PMs) in a way that optimizes the utilization of available resources (CPU, memory, and bandwidth) while reducing energy consumption and preventing overloading is known as virtual machine placement. Conventional VM placement techniques, like static or heuristic-based algorithms, frequently fall short in responding to changing resource needs and dynamic, unpredictable workloads, which results in wasteful resource use, higher operating expenses, etc.

The challenge is to use the Q-learning algorithm, a reinforcement learning technique that permits dynamic and adaptive decision-making, to build and develop an effective framework for virtual machine deployment in cloud data centers. By continuously engaging with the environment, adjusting to shifts in workload patterns, and optimizing for a variety of goals, the framework should be able to learn the best virtual machine placement rules on its own including:-

(i) Resource Utilization: The proposed framework ensures balanced usage of PM by allocating VMs in a way that avoids under utilization and overloading of resources like CPU, memory, and bandwidth.

(ii) Reduction of Computational Costs: It investigates techniques for lowering the overall operational costs of data centers, including power, cooling and hardware maintenance, through optimized VM placement strategies.

(iii) Load Balancing: It prevents resource hotspots by distributing workloads evenly across available PMs, thereby improving performance and avoiding bottlenecks.

(iv) SLA Compliance: It reduces SLA violations by ensuring that VMs are allocated enough resources to meet their performance requirements and quality of service expectations.

(v) Scalability: It handles large-scale data center environments with varying numbers of VMs and PMs, and adapting to fluctuations in demand.

(vi) Adaptability: It responds in real-time to changes in the cloud environment, such as increasing or decreasing workloads, hardware failures, and heterogeneous infrastructure resources.

The proposed framework will automatically learn and improve VM placement policies over time by utilizing the Q-learning algorithm. By using Q-learning, the system can evaluate the state of the cloud environment at any given moment, take the necessary action (assigning virtual machines to specific PMs, for example), receive feedback (in the form of rewards), and modify its placement strategy accordingly. This learning-based approach will increase overall effectiveness by enabling the system to adapt to changing resource requirements and make better decisions. The primary objective is to develop and implement a self-learning, efficient framework that use Q-learning to dynamically optimize the placement of virtual machines in cloud data centers. The framework will increase resource utilization, restrict energy usage, ensure SLA compliance, and scale efficiently to large and dynamic cloud settings to improve cloud infrastructure management performance and reduce operational expenses.

3. State of the Art VM Placement Techniques

For cloud data center virtual machine placement optimization, there are a number of traditional and machine learning-based techniques that show notable variations in effectiveness, precision, and flexibility. While machine learning models use data-driven methodologies to optimize virtual machine placement dynamically, traditional algorithms frequently rely on heuristic or metaheuristic approaches.

(a) Traditional Scheduling Algorithms: Resource allocation in cloud data centers can be simulated by using traditional scheduling techniques for virtual machine placement. Each of these algorithms offers a fundamental method for assigning VMs to hosts in the simulation environment.

(i) First-Come-First-Serve (FCFS): VMs are installed on hosts using FCFS in the order that they arrive. The first VM in the queue will be assigned by the simulation environment to the first host that can meet its resource needs. The next virtual machine in the queue is taken into consideration after a host has been chosen, and this process is repeated until all VMs have been put [2][9][10].

Advantages: It's very simple to implement and requires minimal modifications to the existing VM allocation approaches.

Disadvantages: FCFS can lead to inefficient resource utilization similar to real-world scenarios, as it doesn't consider the workload or resources available on each host beyond basic availability checks.

(ii) Shortest Job First (SJF): SJF assigns VMs according to resource requirements or predicted execution times. In order to maximize host utilization, virtual machines are arranged so that the smallest jobs (based on execution time or resource requirements) are allocated to hosts first [2][11].

Advantages: On varying resource demands, SJF can reduce waiting time and increase resource utilization by allocating smaller VMs first.

Disadvantages: SJF can lead to starvation for larger VMs in real-world scenarios, particularly if the workload contains a continuous stream of short jobs.

(iii) Round Robin (RR): VMs are cyclically assigned to hosts via Round Robin. Without taking into account each host's load or VM requirements, each virtual machine is installed on a host in a sequential order, distributing the load equally among hosts. The allocation rounds back to the initial host after each host has been given a virtual machine [2][12].

Advantages: It provides a simple, fair allocation strategy, preventing any single host from being overloaded. In simulations, it ensures that all hosts receive equal opportunities for VM placement.

Disadvantages: It does not account for each VM's specific resource demands, potentially leading to inefficiencies. This may result in some hosts being underutilized or overburdened depending on VM requirements.

(b) Heuristic and Metaheuristic Algorithms: Heuristic and metaheuristic approaches are popular for VM placement due to their balance of solution quality and computational efficiency, especially in large, complex cloud environments.

(i) Genetic Algorithm (GA): It simulate evolutionary processes to evolve high-quality solutions by selecting, crossing over, and mutating "chromosomes" (candidate solutions) for VM placement. It efficiently balances resource usage, energy consumption and VM migration costs [13].

(ii) Particle Swarm Optimization (PSO): Inspired by social behavior in nature, PSO uses "particles" (solutions) that adjust based on individual experience and the experience of neighboring particles. This algorithm is suitable for load balancing and optimizing energy efficiency [14].

(iii) Ant Colony Optimization (ACO): ACO mimics ant foraging behavior, where ants leave pheromones to guide others to the shortest paths, aiming to find optimal placements through pheromone trails. It reduces resource fragmentation and improves resource utilization [15].

(c) Machine Learning-Based Algorithms: Machine learning algorithms can adapt VM placement strategies based on learned patterns, making them particularly effective in dynamic cloud environments.

(i) Supervised/unsupervised Learning Models: Supervised learning models predict optimal host placements for VMs based on historical data, using features such as resource requirements and host availability. Its effective in environments with predictable workloads [16] while on the other hand unsupervised learning models analyze patterns in data , such as CPU and memory usage to identify optimal VM placements [17].

(ii) Reinforcement Learning (RL): RL algorithms like Q-learning algorithm learn the best placement policy by rewarding positive outcomes (e.g., high resource utilization) and penalizing negative ones (e.g., SLA violations). Its well-suited for energy optimization and adaptive load balancing [18].

The important factor load balancing well presented in [19] which is based on prediction based multi-objective VM placement framework. The cyber security technique using artificial intelligence (AI) is clearly explained in [20] which states the importance of AI in the VM placement techniques. How to use encryption tools in public cloud data servers, that is elaborated in [21]. The cloud security issues explained in [22][23] for better resource management in cloud data centers.

In nutshell, these cutting-edge virtual machine placement strategies enhance resource management capabilities in cloud environments by fusing algorithmic advancements with pragmatic objectives. All of the methods are designed to balance particular goals, such as load balancing, computational cost reduction, energy conservation, and performance enhancement. The related reasearch outcomes have been outlined in table 1.

Table 1 Previous VM Placement related Research Contributions.

Authors	Methods used	Findings	Limitations
Hidayat [24]	Hybrid model: MDP, GA, RF algorithms for VM migration estimation	Hybrid model achieves 99% accuracy in VM migration.	Further exploration of deep learning approaches is recommended and implementation in real-world data centers needs investigation.
Muhaammad [25]	Extreme Learning Machine Prediction-Based Triggering Method	This method improves prediction accuracy and reduces SLA penalties and achieved 10.5% cost function improvement over benchmark approach.	Existing methods over-forecast and have long linear trends and less optimal solutions due to prediction inaccuracies.
Paneshar [26]	Various migration techniques leveraging ML heuristics	Analysis of ML heuristics for VM distribution techniques.	It requires more computational resources and training data.
Jumnal [27]	Fuzzy-based SARSA reinforcement learning algorithm	It reduces energy consumption and resource wastage compared to other algorithms.	Handling fluctuating workload situations while ensuring QoS demands.

Ghasemi [28]	Machine learning-based strategy for VM placement	Improved load balance by 25%, 34%, and 32% for processor, memory, bandwidth.	Load balancing, energy consumption, resource utilization, response time challenges.
Pahlevan [29]	Heuristic & Machine learning based VM allocation method	ML approach improves network traffic by up to 24%.	Traditional approaches lack flexibility for large-scale scenarios.
Farikha [30]	Decision Tree & KNeighborsClassifier for adding new controllers.	Achieved 83% accuracy using Decision Tree and Kneighbors Classifier.	Traditional firewalls cannot examine content inside packets in Cloud Computing.
Wang [31]	Gradient Search and Bin Packing	Utilized gradient search and bin packing for dynamic VM consolidation in multi-tier web applications.	Limited applicability to single-web application setups
Saxena [32]	Whale Optimization Genetic Algorithm (WOGA)	Increased resource utilization up to 30.21%.	Inefficient VM Placement leads to resource wastage and power consumption.
Sui [33]	K-means clustering & Load forecasting algorithm based on genetic algorithm	It lowers energy usage by 49.13% and virtual machine migration by 94.5%.	Occurrence of local optimal solutions during clustering center selection
Chhabra [34]	Optimal VM Placement for Load Balancing (OPLB) method	OPLB improves traffic scalability by up to 49.54%.	It does not compare traditional and machine learning-based algorithms for VM placement.
Dubey [35]	VM allocation approach that is based on water drop method	Better resource utilization, reduced energy usage, increased efficiency, and better overall performance	It is necessary to assess the quality of service offered by the suggested approach for various SLAs and virtual machine instances utilizing meta heuristic-based optimization approaches.
Mejahed [36]	Hybrid approach based on PSO and flower pollination optimization	The recommended method fared better in the simulated studies than the best fit bin packing strategy.	When putting virtual machines, additional factors need to be taken into account in addition to load balancing, live migration, and cost reduction.
Nikzad [37]	Mulit-objective ant colony optimization	The proposed technique reduces energy consumption by 10.3%, SLA violation by 5.3%, and VM migration by 12.5%.	It must be tested using open source tools like OpenStack in a real-world cloud environment.
Radi [38]	VM consolidation method using modified	The results of the investigation show that the recommended approach outperforms the others in terms of energy consumption, SLA violations, and the overall number of virtual machine migrations.	It only draws attention to CPU-intensive tasks. It is necessary to test the suggested method in an actual cloud environment.
Barthwal [39]	Ant Colony Optimization	The recommended approach uses less energy while still adhering to SLAs than the power-aware best fit solution.	Performance can be further improved by making use of additional resources, such as memory, storage, and bandwidth.
Ibrahim [40]	Efficient Adaptive Migration Algorithm	The suggested approach works better in terms of resource usage, shut down hosts, migrations, and SLA violations.	To optimize resource consumption and reduce SLA violations, a load-balanced, resource-aware virtual machine migration method is required.

Mosa [41]	Genetic Algorithm	It reduces over and under utilization of CPU and memory.	There should be more assessments based on actual workload traces.
Tchana [42]	Dynamic Software Consolidation	Around 40% less energy was used in private clouds, and roughly 40.5% less amount was spent on virtual machines on Amazon EC2.	Coordination between software consolidation on virtual machines (VMs) and VM consolidation on PM is necessary to further optimize power gains.
Shafiq [43]	Load Balancing Algorithm	Implemented the least-loaded algorithm, load balancing, and FFD comparison.	Low performance on servers that are heavily used

4. Leveraging Q-Learning for VM Placement

Q-learning [44][45] is a reinforcement learning technique used for decision making in dynamic and uncertain environments. An agent learns a policy for choosing actions in Q-learning by interacting with its surroundings. The agent keeps track of Q-values, which indicate the anticipated cumulative rewards for performing particular actions in specified states, in a Q-table (also known as a Q-function). In order to maximize the cumulative reward over time, the agent learns to adjust Q-values based on the observed rewards through trial and error.

Numerous decision-making tasks, such as game play, robotic control, and effective resource allocation in cloud data centers, have seen the successful use of Q-learning [46]. When it comes to task scheduling, Q-learning can figure out the best way to allocate tasks to computer resources based on the system's current condition, the nature of the tasks, and external variables.

The Q-table, which is represented as Q(s, a) where s is the state and a is the action, uses a state-activity pair to index a Q value as a cumulative reward. The reward/penalty of a particular state-action pair determines how the Q-table is dynamically updated.

$$Q_{(t+1)}(s_t + a_t) = Q(s_t + a_t) + a_t(r_t + c\max(Q_t(s_{t+1} + a_{t+1})) - Q(s_t + a_t)) \qquad (1)$$

where r is reward/penalty, α is the learning rate within [0,1], and c is the discount factor within [0,1]. These values are computed as follows:

$$\alpha = 1 - 0.9 * \frac{t}{\text{MaxIteration}} \qquad (2)$$

Exploration or exploitation is indicated by the value of the parameter α. The newly obtained data are given a higher priority if the value is near 1, which indicates that all defined states are explored. On the other hand, current data are prioritized for exploitation if the value is near zero. The value of parameter c, which indicates whether to accept the prior award or penalty or the present one, was set to 0.

The value of parameter r is set as follows:

$r_t = 1$, if the current action improves the solution

$r_t = -1,$ otherwise
(3)

Algorithm 1: The Q-learning algorithm

```
for each state S=[$s_1$,$s_2$,…,$s_n$] and action A=[$a_1$,$a_2$,…,$a_n$] do
    set $Q(s_t + a_t) = 0$
end
Randomly select an initial state, $s_t$
while stopping criteria are not met do
    Select the best action $a_t$, for the current state $s_t$ from Q-table
    Execute action $a_t$ and get immediate reward/penaulty $r_t$ using Eq (3)
    Get the maximum Q value for the next state $s_{t+1}$
    Update $\alpha$ using Eq (2)
    Update Q-table entry using Eq (1)
    Update the current state $s_t$=$s_{t+1}$
End
return the updated Q(s,a) table
```

The architecture depicted in the fig.1 is a high-level design for a cloud-based task scheduling and resource management system, utilizing a reinforcement learning (RL) approach specifically employing the Q-learning algorithm. In order to ensure effective resource usage, minimize the overall Makespan-the amount of time needed to finish all submitted tasks-and lower operating costs, this architecture is set up to optimize task scheduling across a cloud infrastructure in comparison to traditional models using cloud/fog computing [47].

The RL(Q) Scheduler, which uses Q-learning RL algorithms, is the central component of this system. In cloud environments, the algorithm is designed to manage complicated and dynamic job scheduling circumstances. A model-free reinforcement learning method called Q-learning learns a policy to maximize the cumulative reward by acting in the way that best suits the current situation. The System Access Interface, which acts as the system's access point, is how users submit tasks. The Task Manager, a key component in-charge of managing available resources, keeping an eye on task statuses, and distributing the load throughout the system, receives these tasks after that. The Task Manager has a Task Monitor to monitor how tasks are being completed, a Resource Monitor to control the computational resources that are available, and a Load Balancer to divide duties evenly across these resources so that no one is overworked.

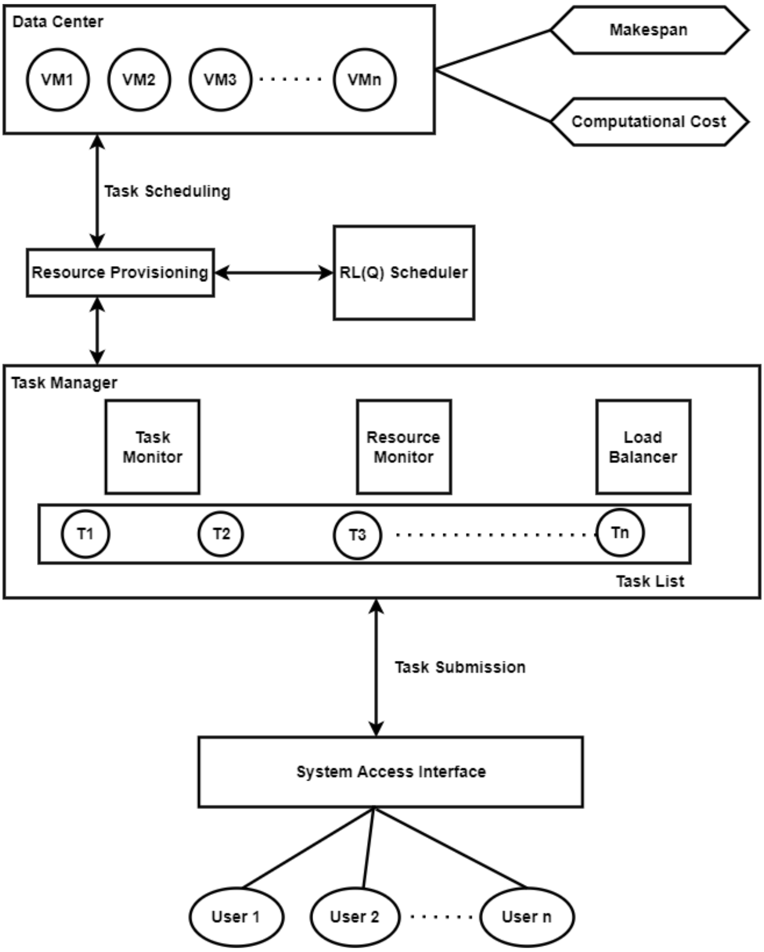


Fig. 1 Efficient Framework for Resource Utilization and Computational Cost Reduction.

The Resource Provisioning module, which works closely with the RL(Q) Scheduler, is in-charge of assigning the right resources, like VMs in a data center, to the tasks in accordance with the scheduler's choices. This interaction is essential because it enables real-time resource allocation modifications, guaranteeing that tasks are carried out effectively across all available virtual machines (VM1, VM2, …, VMn).

The Data Center is where tasks are carried out after they have been planned and resources have been allocated. The Makespan, which is the total amount of time needed to do all the activities, is used to gauge how well this entire process is performing. The main objective of the system is to reduce the Makespan in order to increase overall efficiency while lowering computational costs.

This architecture emphasizes the value of reinforcement learning in optimizing resource management and task execution in cloud computing settings, and it provides a sophisticated approach to cloud task scheduling through the integration of the Q-learning algorithm. By cutting down on job completion times, this solution not only improves resource utilization but also lowers computational expenses and boosts cloud service responsiveness.

Components:

(i) System Access Interface:

o This is the entry point for all users (User 1, User 2, …, User n) to submit tasks to the system. The interface collects and forwards tasks to the Task Manager.

(ii) Task Manager:

o Task Monitor: It monitors the status of tasks that have been submitted.

o Resource Monitor: It keeps track of the available resources in the system.

o Load Balancer: It distributes tasks evenly across resources to ensure optimal performance.

o Task List: It represents the collection of tasks (T1, T2, T3, …, Tn) that are currently managed by the Task Manager.

(iii) RL(Q) Scheduler:

o This component optimizes task scheduling through the use of a hybrid reinforcement learning approach. To distribute tasks effectively, it communicates with the Task Manager and Resource Provisioning.

(iv) Resource Provisioning:

o It allocates resources (e.g., VMs) based on the task's requirements and the scheduling decisions made by the RL(Q) Scheduler.

(v) Data Center:

o It contains Virtual Machines (VM1, VM2, …, VMn) that execute the scheduled tasks. The VMs are managed and provisioned based on the scheduling and resource provisioning decisions.

(vi) Makespan:

o It represents the total time taken to complete all the tasks. It is an important metric for evaluating the efficiency of the scheduling algorithm.

(vii) Computational Cost:

o Computational cost represents the reduction of operational expenses and resource waste to improve efficiency and responsiveness.

Data Flow:

(i) Task Submission:

o Users submit tasks through the System Access Interface, which then forwards them to the Task Manager.

(ii) Task Management:

o The Task Manager monitors tasks, resources, and load balancing, ensuring that tasks are ready to be scheduled.

(iii) Scheduling and Resource Provisioning:

o The RL(Q) Scheduler interacts with Resource Provisioning to allocate the necessary resources for each task.

o Resource Provisioning then allocates the required resources (e.g., VMs) within the Data Center.

(iv) Task Execution:

o Tasks are executed on the allocated VMs in the Data Center. The execution time contributes to the overall Makespan.

(v) Makespan Calculation:

o The system monitors the completion of tasks to calculate the Makespan, which is used to assess the performance of the scheduling process.

(vi) Computational Cost Calculation:

o The simulation system calculates the average computational cost by applying various techniques from traditional FCFS to Q-learning algorithm.

5. Simulation Setup

CloudSim is an open-source Java-based framework and toolset to model and simulate cloud computing applications and infrastructures. It was originally developed at the University of Melbourne's Cloud Computing and Distributed Systems (CLOUDS) Laboratory. Users may model and simulate workloads, data centers, virtual machines, and more with CloudSim. It also provides basic classes for characterizing computing resources, rules, applications, and people. A Java-based Integrated Development Environment (IDE), like Eclipse, is set up for the experiment. The toolkit includes multiple packages with Java code for different implementations.

The deployment of virtual machines in the cloud is essential for optimizing resource allocation, reducing computational cost, and improving overall performance. It can also model various scenarios for virtual machine setup in a cloud environment to evaluate the impact on QoS measures like resource utilization, computational cost etc. By analyzing various techniques and rules for the deployment of virtual machines, it helps to create better and more efficient operations. Thus, by simulating different scenarios, CloudSim helps researchers and practitioners analyze how choices regarding the location of virtual machines impact computational cost, latency, energy consumption, and resource usage in cloud computing environments.

The simulation demonstrates the efficacy and efficiency of the suggested method in resolving computational issues such as job distribution, resource optimization, computational cost reduction, and latency reduction. The following simulation environment configured for finding results:-

- No. of data centres: There is 3-20 data center used in this simulation.
- No. of broker: 1.
- No. of hosts: The data center contains 2 hosts (physical machines).
- No. of CPUs: Each host has 1 CPU.
- No. of virtual machines: 10-1000.
- No. of tasks (cloudlets): Varies from 100 to 1000.
- Population size (No. of particles): 25
- MIPS of CPU per virtual machine (Millions of Instructions Per Second):

Homogenous: Each VM has a MIPS value of 1000-1200.

Heterogeneous: VMs have varying MIPS values greater than 2000 but less than or equal to 20000.

- RAM: Each VM is allocated 512 -16384 MB of RAM.

These parameters define the computational resources and network characteristics of the simulated environment, influencing the performance and behavior of the virtual machines during the simulation. The variability in the number of VMs and their heterogeneity in terms of RAM and MIPS allows for a range of scenarios to be tested, which can help in understanding the performance and scalability of the data center under different conditions.

6. Results and Findings

During the tests, the simulator acts like a real-life deployment environment where Data center, tasks (Cloudlets), and VMs are very different from one another. This lets us see how well and quickly cloud computing solutions work at solving modern computing problems like lowering latency, making the best use of resources, lowering computational costs, and distributing tasks evenly.

Result shows that machine learning algorithms specially Q-learning algorithm outperforms in reducing computational costs, and improving resource utilization, and improving overall system performance while comparing with traditional, heuristics and metaheuristics algorithms.

6.1 Makespan:

In cloud computing, makespan refers to the time taken to execute a batch of tasks on multiple VMs or servers. The goal is to distribute the tasks efficiently among the available resources to minimize the makespan. The table 2 showing performance of algorithms where Q-learning algorithm performance is so high on varying workloads (tasks) from 100 to 500 and while increasing the number of VMs from 100 to 1000 using Q-learning algorithm, the performance of algorithm is again increasing, on various workloads that is shown in table 3.

Table 2: Makespan results of various algorithms on increasing workloads.

Algorithm/Cloudlet	100	200	300	400	500
FCFS	9656.54	9797.40	12306.25	9119.48	9227.61
RR	9520.14	9702.03	10069.51	11264.45	15892.50
SJF	8960.23	9526.12	9085.38	12376.41	17183.32
PSO	6470.50	7898.12	8642.94	9467.24	10308.48
Q-learning	478.82	503.74	651.60	810.14	920.91

Table 3: Makespan results of Q-learning algorithm on increasing VMs.

(a) Cloudlet: 100

VM	Makspan
100	68
250	25
500	12.09
1000	12.09

(b) Cloudlet: 500

VM	Makspan
100	180.2
250	110
500	71
1000	32

(c) Cloudlet: 1000

VM	Makspan
100	350
250	190
500	85
1000	69

As makespan represents the time taken to execute the batch of task that is so reduced using Q-learning on increasing workloads and increasing number of VMs as well so the Q-learning algorithm is suitable to produce high resource utilization about 92% in comparison to traditional algorithms.

6.2 Computational Cost:

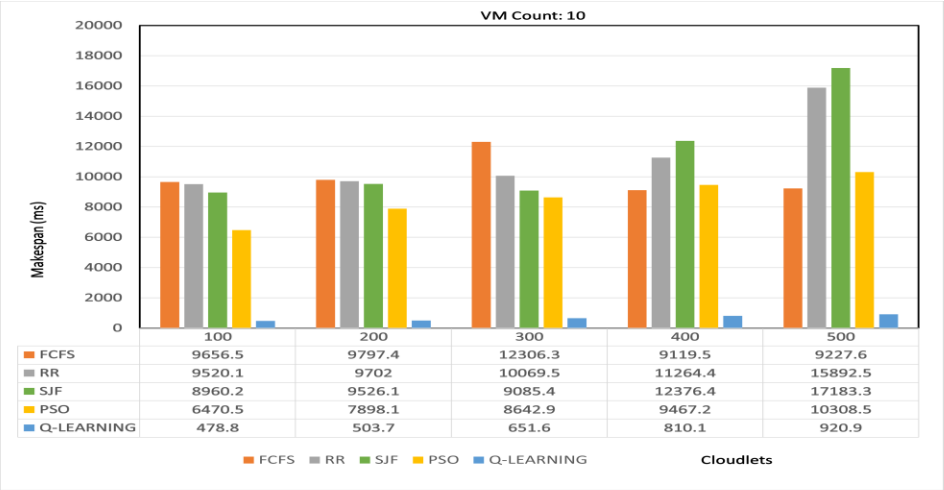
Unlike static policies, which may frequently migrate VMs in response to the changing demands, Q-learning minimizes unnecessary migrations which not only saves computational resources but also reduces computational cost. Table 4 shows the comaparion result of avearge computational cost on varying workloads. Average computational cost is reduced about 34% using Q-learning algorithm in comparison to traditional algorithms.

Table 4: Average computational cost results of various algorithms on increasing workloads.

Cloudlet	100	200	300	400	500
FCFS	1870.45	1766.97	1145.38	842.24	715.60
RR	1790.86	1696.49	1209.79	896.30	670.56
SJF	1840.23	1755.68	1158.51	834.38	690.76
PSO	1410.25	1202.80	814.40	655.79	493.63
Q-learning	910.12	771.45	551.60	421.58	325.27

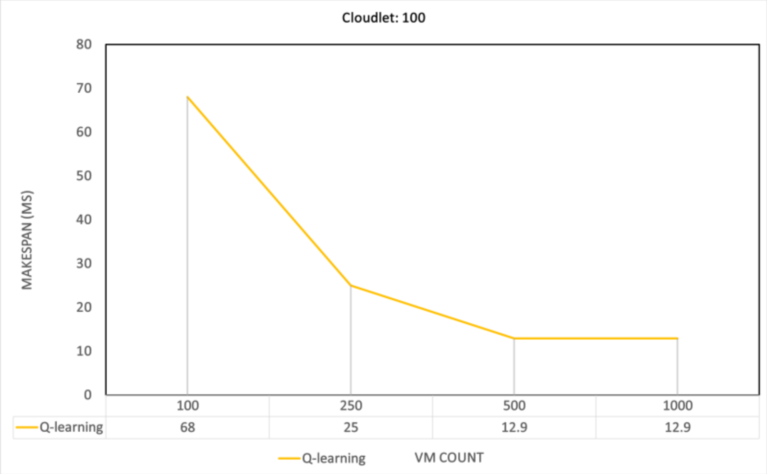
7. Conclusion

In conclusion, applying a Q-learning algorithm for VM placement in cloud data centers presents a promising approach for enhancing resource allocation, optimizing performance, and reducing computational costs. By employing a reinforcement learning technique, Q-learning continuously learns and adapts to the dynamic nature of cloud environments, efficiently handling the high variability in demand and resource availability. This leads to more intelligent decision-making in VM placement, improving load balancing, minimizing latency, and maximizing resource utilization. The graph 1 shows that on fixed number of VM, the makespan is so decreased in comparison to other algorithms and graph 2 shows that on increasing order of VMs and fix workloads, the again makespan is decreasing that shows the efficient resource utilization about 92% in comparison to traditional algorithms.



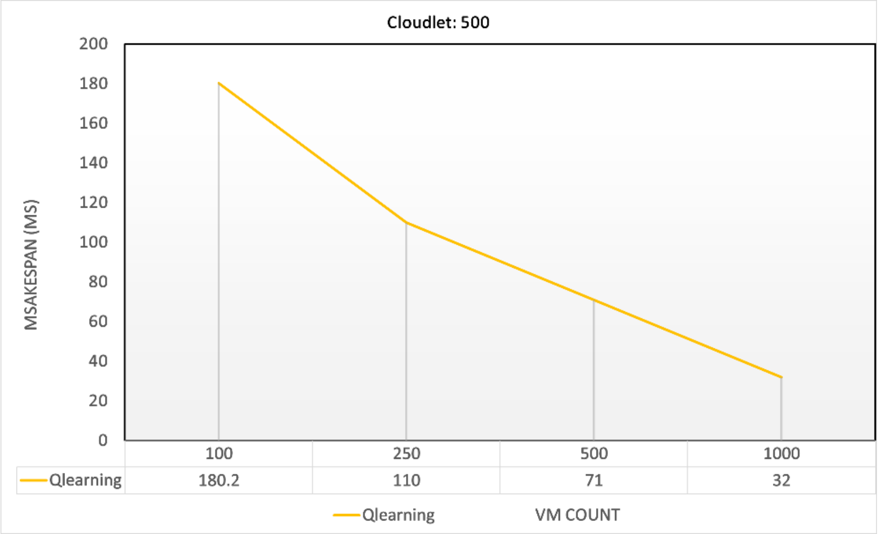
	100	200	300	400	500
FCFS	9656.5	9797.4	12306.3	9119.5	9227.6
RR	9520.1	9702	10069.5	11264.4	15892.5
SJF	8960.2	9526.1	9085.4	12376.4	17183.3
PSO	6470.5	7898.1	8642.9	9467.2	10308.5
Q-LEARNING	478.8	503.7	651.6	810.1	920.9

Graph 1 Makespan Results of various algorithms on fix VM and varying workloads.

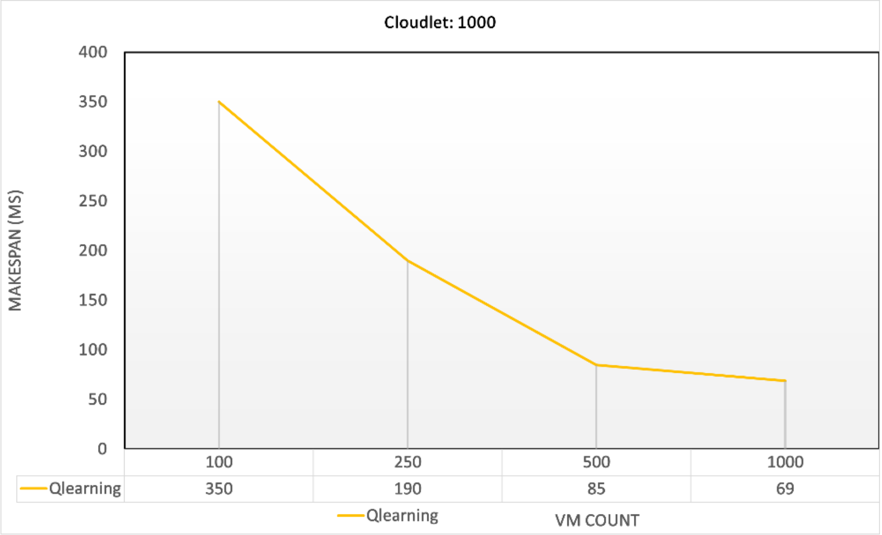


	100	250	500	1000
Q-learning	68	25	12.9	12.9

(a)



(b)

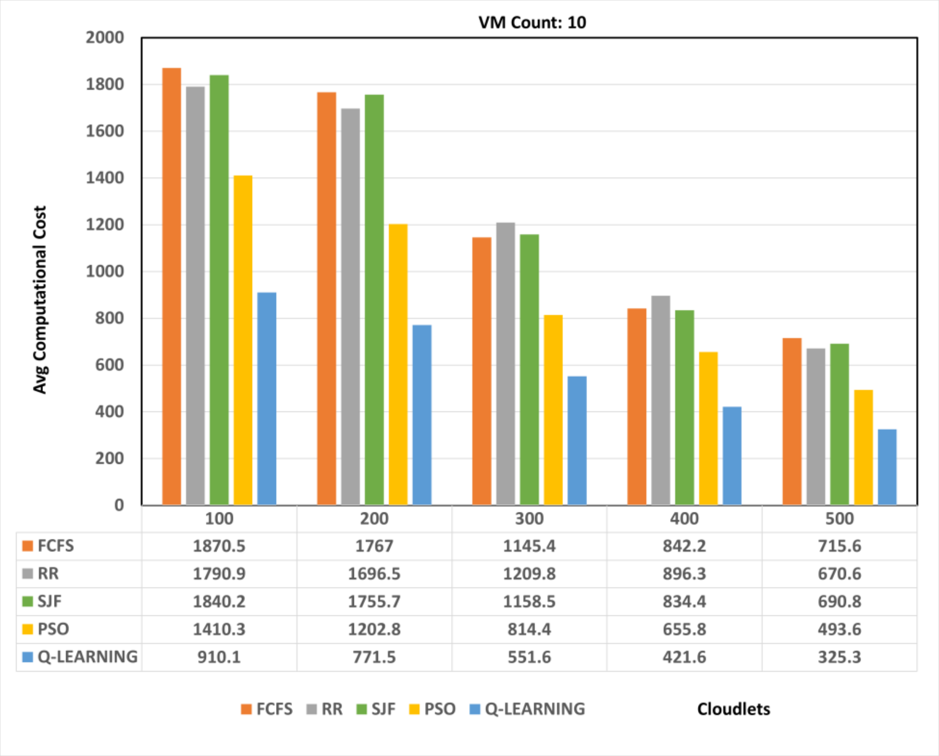


(c)

Graph 2 (a),(b),(c) Result of Q-learning on fix workload and varying VMs.

The Graph 3 shows the results of average (avg) computational cost of various algorithms on increasing workloads where Q-learning resulting minimum and better avg computational cost about 34% in comparison to traditional algorithms.

Compared to traditional approaches, which often rely on static rules or predefined policies, Q-learning offers flexibility and adaptability. It can respond to changing conditions in real time, enabling a more agile management of resources. The algorithm's ability to explore and exploit various placement strategies over time helps in converging toward optimal or near-optimal solutions. However, the initial training period may require computational resources and time to reach a stable and effective policy.



	100	200	300	400	500
FCFS	1870.5	1767	1145.4	842.2	715.6
RR	1790.9	1696.5	1209.8	896.3	670.6
SJF	1840.2	1755.7	1158.5	834.4	690.8
PSO	1410.3	1202.8	814.4	655.8	493.6
Q-LEARNING	910.1	771.5	551.6	421.6	325.3

Graph 3 Avg Computational Cost Results of various algorithms on fix VM and varying workloads.

Overall, Q-learning offers a robust, scalable solution that addresses the limitations of conventional placement methods, particularly in large, dynamic cloud data centers. Future work can focus on integrating other advanced machine learning techniques with Q-learning to further enhance its efficiency and effectiveness in VM placement, such as by incorporating neural networks for state approximation or employing multi-agent systems for cooperative learning across distributed data centers. This approach can drive further innovation in cloud infrastructure management, improving service quality and cost-efficiency for cloud providers and users alike.

Acknowledgement

We express our gratitude to the Research Department of the University for providing the manuscript number IU/R&D/2024-MCN0003156 in accordance with their criteria. This identification makes it simpler to track & communicate about our research throughout the publication process. We also like to thank each & everyone who supported in this research work.

References

1. Duong-Ba, T., Tran, T., Nguyen, T., & Bose, B. (2021). A Dynamic Virtual Machine

Placement and Migration Scheme for Data Centers. IEEE Transactions on Services Computing, 14(02), 329-341. (Journal Article)

2. Ponraj, A. (2019). Optimistic virtual machine placement in cloud data centers using queuing approach. Future Generation Computer Systems, 93, 338-344. (Journal Article)
3. Toumi, N., Bagaa, M., & Ksentini, A. (2023). Machine learning for service migration: a survey. IEEE Communications Surveys & Tutorials, 25(3), 1991-2020. (Journal Article)
4. Long, S., Li, Z., Xing, Y., Tian, S., Li, D., & Yu, R. (2020, December). A reinforcement learning-based virtual machine placement strategy in cloud data centers. In 2020 IEEE 22nd International Conference on High Performance Computing and Communications; IEEE 18th International Conference on Smart City; IEEE 6th International Conference on Data Science and Systems (HPCC/SmartCity/DSS) (pp. 223-230). IEEE. (Conference Article)
5. Jena, U. K., Das, P. K., & Kabat, M. R. (2022). Hybridization of meta-heuristic algorithm for load balancing in cloud computing environment. Journal of King Saud University-Computer and Information Sciences, 34(6), 2332-2342. (Journal Article)
6. Deep Singh, K., & Sood, S. K. (2020). 5G ready optical fog-assisted cyber-physical system for IoT applications. IET Cyber-Physical Systems: Theory & Applications, 5(2), 137-144. (Journal Article)
7. Sood, S. K., & Deep Singh, K. (2023). Hmm-based secure framework for optical fog devices in the optical fog/cloud network. Journal of Optical Communications, 44(4), 475-483. (Journal Article)
8. Yucel, S. (2023, July). Reinforcement Learning Approach to Server Selection and Load Balancing for Collaborative Virtual Services. In 2023 Congress in Computer Science, Computer Engineering, & Applied Computing (CSCE) (pp. 1206-1213). IEEE. (Journal Article)
9. Masdari, M., Nabavi, S. S., & Ahmadi, V. (2016). An overview of virtual machine placement schemes in cloud computing. Journal of Network and Computer Applications, 66, 106-127. (Journal Article)
10. Xu, M., Tian, W., & Buyya, R. (2017). A survey on load balancing algorithms for virtual machines placement in cloud computing. Concurrency and Computation: Practice and Experience, 29(12), e4123. (Journal Article)
11. Mikram, H., El Kafhali, S., & Saadi, Y. (2022, October). Processing time performance analysis of scheduling algorithms for virtual machines placement in cloud computing environment. In International Conference On Big Data and Internet of Things (pp. 200-211). Cham: Springer International Publishing. (Conference Article)
12. Panda, S. K., & Jana, P. K. (2017). An efficient request-based virtual machine placement algorithm for cloud computing. In Distributed Computing and Internet Technology: 13th International Conference, ICDCIT 2017, Bhubaneswar, India, January 13-16, 2017, Proceedings 13 (pp. 129-143). Springer International Publishing. (Conference Article)
13. Lu, J., Zhao, W., Zhu, H., Li, J., Cheng, Z., & Xiao, G. (2022). Optimal machine placement based on improved genetic algorithm in cloud computing. The Journal of Supercomputing, 1-29. (Journal Article)
14. Nagle, M., & Nagar, V. (2024). Cloud Computing: Optimization using Particle Swarm Optimization to Improve Performance of Cloud. EAI Endorsed Transactions on Internet of Things, 10. (Journal Article)
15. Peake, J., Amos, M., Costen, N., Masala, G., & Lloyd, H. (2022). PACO-VMP: parallel ant colony optimization for virtual machine placement. Future Generation Computer Systems, 129, 174-186. (Journal Article)
16. Talwani, S., Singla, J., Mathur, G., Malik, N., Jhanjhi, N. Z., Masud, M., & Aljahdali, S. (2022). Machine-learning-based approach for virtual machine allocation and migration. Electronics, 11(19), 3249. (Journal Article)

17. Khan, W., & Haroon, M. (2022). An unsupervised deep learning ensemble model for anomaly detection in static attributed social networks. International Journal of Cognitive Computing in Engineering, 3, 153-160. (Journal Article)
18. Xu, H., & Jian, C. (2024). A meta reinforcement learning-based virtual machine placement algorithm in mobile edge computing. Cluster Computing, 27(2), 1883-1896. (Journal Article)
19. Saxena, D., Singh, A. K., & Buyya, R. (2021). OP-MLB: an online VM prediction-based multi-objective load balancing framework for resource management at cloud data center. IEEE Transactions on Cloud Computing, 10(4), 2804-2816. (Journal Article)
20. Haroon, M., Afsaruddin Khan, M. M., Tripathi, S. A., & Ahmad, K. (2024). Cyber-security technique for profound analytics of big data using AI. Computational Intelligence Applications in Cyber Security, 247. (Journal Article)
21. Daman, R., & Tripathi, M. M. (2015). Encryption tools for secured health data in public cloud. International Journal of Innovative Science, Engineering & Technology, 2(11), 843-848. (Journal Article)
22. Khan, A. M., Ahmad, S., & Haroon, M. (2015, April). A comparative study of trends in security in cloud computing. In 2015 Fifth International Conference on Communication Systems and Network Technologies (pp. 586-590). IEEE. (Conference Article)
23. Ahmad, S., Beg, M. R., Ahmad, J., & Barua, N. (2010). Meet In The Middle Attack: A Cryptanalysis Approach. International Journal of Computer Applications, 975(1), 1-7, doi: 10.5120/467-772. (Journal Article)
24. Hidayat, T., Ramli, K., Thereza, N., Daulay, A., Rushendra, R., & Mahardiko, R. (2024, July). Machine Learning to Estimate Workload and Balance Resources with Live Migration and VM Placement. In Informatics (Vol. 11, No. 3, p. 50). MDPI. (Journal Article)
25. Muhaammad, N. M., Ambursa, F. U., & Faruk, N. (2023). Two-Phase Virtual Machine Placement in Cloud Computing Data Centers Using Extreme Learning Machine Prediction-Based Triggering Method. (Journal Article)
26. Panesar, G. S., & Chadha, R. (2023, November). Migration Techniques for Virtual Machine Distribution Using Machine Learning Heuristics Application in Cloud Computing. In 2023 3rd International Conference on Technological Advancements in Computational Sciences (ICTACS) (pp. 1486-1493). IEEE. (Conference Article)
27. Jumnal, A., & Kumar, S. D. (2021, February). Optimal vm placement approach using fuzzy reinforcement learning for cloud data centers. In 2021 Third International Conference on Intelligent Communication Technologies and Virtual Mobile Networks (ICICV) (pp. 29-35). IEEE. (Conference Article)
28. Ghasemi, A., & Toroghi Haghighat, A. (2020). A multi-objective load balancing algorithm for virtual machine placement in cloud data centers based on machine learning. Computing, 102(9), 2049-2072. (Journal Article)
29. Pahlevan, A., Qu, X., Zapater, M., & Atienza, D. (2017). Integrating heuristic and machine-learning methods for efficient virtual machine allocation in data centers. IEEE transactions on computer-aided design of integrated circuits and systems, 37(8), 1667-1680. (Journal Article)
30. Frikha, H., Meddeb-Makhoulf, A., & Zarai, F. (2023). Allocation of virtual machine in a cloud environment based on machine learning. (Journal Article)
31. Wang, P., Nagrecha, K., & Vasconcelos, N. (2021). Gradient-based algorithms for machine teaching. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 1387-1396). (Conference Article)
32. Saxena, D., Gupta, I., Kumar, J., Singh, A. K., & Wen, X. (2021). A secure and multiobjective virtual machine placement framework for cloud data center. IEEE Systems Journal, 16(2), 3163-3174. (Journal Article)
33. Sui, X., Liu, D., Li, L., Wang, H., & Yang, H. (2019). Virtual machine scheduling strategy based on machine learning algorithms for load balancing. EURASIP Journal on Wireless

Communications and Networking, 2019(1), 160. (Journal Article)

34. Chhabra, S., & Singh, A. K. (2019, June). Optimal VM placement model for load balancing in cloud data centers. In 2019 7th International Conference on Smart Computing & Communications (ICSCC) (pp. 1-5). IEEE. (Conference Article)
35. Dubey, K., & Sharma, S. C. (2022). An extended intelligent water drop approach for efficient VM allocation in secure cloud computing framework. Journal of King Saud University-Computer and Information Sciences, 34(7), 3948-3958, doi: 10.1016/j.jksuci.2020.11.001. (Journal Article)
36. Mejahed, S., & Elshrkawey, M. (2022). A multi-objective algorithm for virtual machine placement in cloud environments using a hybrid of particle swarm optimization and flower pollination optimization. PeerJ Computer Science, 8, e834, doi: 10.7717/PEERJ-CS.834. (Journal Article)
37. Nikzad, B., Barzegar, B., & Motameni, H. (2022). Sla-aware and energy-efficient virtual machine placement and consolidation in heterogeneous DVFS enabled cloud datacenter. IEEE Access, 10, 81787-81804, doi: 10.1109/ACCESS.2022.3196240. (Journal Article)
38. Radi, M., Alwan, A. A., & Gulzar, Y. (2023). Genetic-based virtual machines consolidation strategy with efficient energy consumption in cloud environment. IEEE Access, 11, 48022-48032, doi:10.1109/ACCESS.2023.3276292. (Journal Article)
39. Barthwal, V., & Rauthan, M. M. S. (2021). AntPu: a meta-heuristic approach for energy-efficient and SLA aware management of virtual machines in cloud computing. Memetic Computing, 13, 91-110, doi: 10.1007/s12293-020-00320-7. (Journal Article)
40. Ibrahim, M., Imran, M., Jamil, F., Lee, Y. J., & Kim, D. H. (2021). EAMA: Efficient adaptive migration algorithm for cloud data centers (CDCs). Symmetry, 13(4), 690, doi: 10.3390/sym13040690. (Journal Article)
41. Mosa, A., & Sakellariou, R. (2019, July). Dynamic virtual machine placement considering CPU and memory resource requirements. In 2019 IEEE 12th international conference on cloud computing (CLOUD) (pp. 196-198). IEEE, doi: 10.1109/CLOUD.2019.00042. (Conference Article)
42. Tchana, A., De Palma, N., Safieddine, I., & Hagimont, D. (2016). Software consolidation as an efficient energy and cost saving solution. Future Generation Computer Systems, 58, 1-12, doi:10.1016/j.future.2015.11.027. (Journal Article)
43. Shafiq, D. A., Jhanjhi, N. Z., & Abdullah, A. (2022). Load balancing techniques in cloud computing environment: A review. Journal of King Saud University-Computer and Information Sciences, 34(7), 3910-3933, doi:10.1016/j.jksuci.2021.02.007. (Journal Article)
44. Ghasemi, A., Toroghi Haghighat, A., & Keshavarzi, A. (2024). Enhancing virtual machine placement efficiency in cloud data centers: a hybrid approach using multi-objective reinforcement learning and clustering strategies. Computing, 106(9), 2897-2922. (Journal Article)
45. Tran, C. H., Bui, T. K., & Pham, T. V. (2022). Virtual machine migration policy for multi-tier application in cloud computing based on Q-learning algorithm. Computing, 104(6), 1285-1306. (Journal Article)
46. Masoumzadeh, S. S., & Hlavacs, H. (2013, October). Integrating VM selection criteria in distributed dynamic VM consolidation using Fuzzy Q-Learning. In Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013) (pp. 332-338). IEEE. (Conference Article)
47. Jaiswal, V. K., Ahmad, J., Singh, K.D. (2024). Leveraging Fog Computing for Virtual Machine Placement in the Cloud Computing Environment. Journal of Electrical Systems, 20(3), 6342-6353. (Journal Article)