International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7179
***Article Received**: 21 April 2023 **Revised:** 09 June 2023 **Accepted:** 28 June 2023*

A Deep Reinforcement Learning-Based Model for Optimal Resource Allocation and Task Scheduling in Cloud Computing

Lamia Alhazmi
Department of Management Information System,
College of Business Administration, Taif University, P.O Box 11099,
Taif 21944, Saudi Arabia.
e-mail: Lamia.s@tu.edu.sa

Abstract—The advent of cloud computing has dramatically altered how information is stored and retrieved. However, the effectiveness and speed of cloud-based applications can be significantly impacted by inefficiencies in the distribution of resources and task scheduling. Such issues have been challenging, but machine and deep learning methods have shown great potential in recent years. This paper suggests a new technique called Deep Q-Networks and Actor-Critic (DQNAC) models that enhance cloud computing efficiency by optimizing resource allocation and task scheduling. We evaluate our approach using a dataset of real-world cloud workload traces and demonstrate that it can significantly improve resource utilization and overall performance compared to traditional approaches. Furthermore, our findings indicate that deep reinforcement learning (DRL)-based methods can be potent and effective for optimizing cloud computing, leading to improved cloud-based application efficiency and flexibility.

Keywords- Learning; optimizing; training; scheduling; planning; reward; computation; experience replay.

I. INTRODUCTION

Cloud computing [1] represents an idea that provides on-demand networked accessibility to shared pooled technological assets such as servers, databases, bandwidth, and storage spaces [2]. Cloud computing allows enterprises to trim the overall cost of their equipment and facilities drastically, thus, increasing or decreasing capacity as required and getting access to many resources/ services, and essential operative applications[3]. Therefore, optimizing resource distribution and task planning is crucial, especially as increasingly more businesses move to the cloud infra.

Resource allocation [4] commonly refers to the allocation/distribution of technological resources, including virtual machines (VM), communication bandwidth on the network, and storage spaces for various activities or operable applications. The intention is to limit resource wastage while providing sufficient resources for each activity or application to run well. Similarly, the primary intent of task scheduling is to fulfil work in a timely and effective way while making the most effective utilization of all available resources [5].

VM consolidation, load balancing, and task scheduling strategies are only some of the approaches that can be employed to optimize resource allocation and task execution planning [6]. The intended effect of load balancing is to prevent any resource from being overburdened while several others remain idle. Consolidating multiple VMs [7] into an all-encompassing hardware host could potentially lower equipment expenses and maximize the use of available resources. Tasks/activities in a long-run operation are scheduled for execution according to an algorithm that considers factors, including their order of priority, expected runtime, and resource needs.

In cloud computing, optimizing the distribution of resources and scheduling tasks may lead to reductions in infrastructure expenses, higher resource utilization, shorter task completion times, and better Quality of Service (QoS) [8]. However, these advantages can only be exploited with a thorough familiarity with the foundational technology and meticulous preparation and implementation.

A. Necessity of Optimizing Resource Allocation and Task Scheduling in Cloud Computing

Figure 1 depicts several factors that make cloud service providers need to optimize resource allocation and task scheduling [9].

Cost savings: The use of cloud computing allows for on-demand access to computer resources, which may result in cost savings; nevertheless, using these resources does incur a cost. Organizations may minimize IT infrastructure expenses by using their computer resources better through effective utilization of resources and task planning.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7179
***Article Received**: 21 April 2023 **Revised:** 09 June 2023 **Accepted:** 28 June 2023*

Improved resource utilization: Many organizations overprovision resources to ensure that they have enough capacity to handle peak loads. However, this can result in wasted resources during periods of low demand. Organizations can improve resource utilization and reduce wastage by optimizing resource allocation and task scheduling.

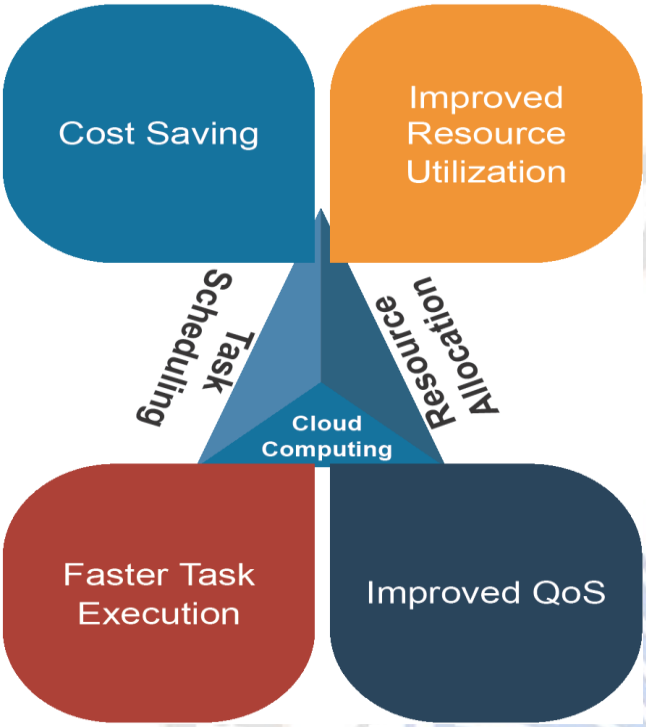

Figure 1. Necessity of Optimization in Cloud Computing

Better performance: Optimizing resource allocation and task scheduling can result in faster task execution times and improved quality of service, which can lead to better performance for applications and services running in the cloud.

Scalability: Cloud computing allows businesses to increase or decrease their computer capacity as required. However, to take full advantage of this flexibility, optimizing resource allocation and task scheduling is necessary to ensure that resources are used efficiently and effectively.

Competitive advantage: In today's competitive business environment, organizations that can operate more efficiently and cost-effectively have a distinct advantage over their competitors. Optimizing resource allocation and task scheduling can help organizations achieve this advantage by reducing costs, improving performance, and enhancing scalability.

B. Problem Definition

An optimization problem is always utilized to describe the context, intending to minimize the cost of performing the tasks while addressing limitations on resource accessibility, resource allocation, load distribution, balancing, and task scheduling. Thus, such a scenario results in the formation of required expressions for the computational issue of optimizing resource allocation and task scheduling in cloud infrastructure:

Axiom 1: Given a set of tasks $T = \{t_1, t_2, t_3, \ldots, t_n\}$ and a set of computing resources $R = \{r_1, r_2, r_3, \ldots, r_n\}$ with different capacities and costs, the objective is to allocate resources to tasks and schedule them in a way that minimizes the cost of executing the tasks while meeting performance and QoS requirements.

Proof: Let T be the set of tasks and R be the set of computing resources. Each resource $r \in R$ has a finite capacity, $c(r)$, which is the maximum amount of work that can be performed on that resource in a given time period, $\boldsymbol{\zeta}$. Each task $t \in T$ requires a certain amount of resources to execute, denoted by $d(t, r)$, where r is the resource on which task t is executed.

Let $X = \{x(t,r) | t\ T, r \in R\}$ be a binary determination factor that denotes whether task t is allocated to resource r.

$$\boldsymbol{x(t,r) = \begin{cases} 1, & t \to r \\ 0, & otherwise \end{cases}} \quad (1)$$

Let $\mathbb{C} = \{c(r) | r \in R\}$ be the capacities of the computing resources, and let $\text{Þ} = \{p(t) | t \in T\}$ be the performance requirements of the tasks.

While still satisfying performance criteria and resource allocation limits, the overall cost of completing the activities must be maintained at the lowest possible level. Therefore, consider a cost function $\Theta(t, r)$ that contains both the resource and performance cost, describing the total cost for performing task t on resource r. Then, this allows us to express the optimization issue as:

$$\boldsymbol{min \sum t \in T \sum r \in R \cdot \mathbb{C}(t,r) \cdot x(t,r)} \quad (2)$$

subject to $Þ(t)$, $\mathbb{C}(r)$, $\delta(a)$, X, which is expressed in equations (3), (4), (5), and (6).

$$\boldsymbol{\sum r \in R \cdot x(t,r) \cdot d(t,r) \geq Þ(t) \ \ \forall t \in T} \quad (3)$$

$$\boldsymbol{\sum t \in T \cdot x(t,r) \cdot d(t,r) \geq \mathbb{C}(t) \ \ \forall r \in R} \quad (4)$$

$$\boldsymbol{\sum t \in T \cdot x(t,r) \leq \delta(a) \ \forall r \in R;\ \delta(a) = 1} \quad (5)$$

$$\boldsymbol{x(t,r) \in \{0,1\} \ \forall t \in T,\ r \in R} \quad (6)$$

The first constraint confirms that all tasks' performance needs are fulfilled, while the second guarantees that all resources' capacity limitations are met. The third constraint mandates that just a single resource be used for each task, while the fourth mandates that the decision variable x(t, r) be binary in nature.

Several methods exist for addressing the optimization issue. The solution offers a cost-effective resource allocation to activities as

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7179
***Article Received**: 21 April 2023 **Revised:** 09 June 2023 **Accepted:** 28 June 2023*

well as a schedule for task execution that considers capacity and performance needs.

C. Scope of the Research Work

Inefficiencies in cloud computing's resource allocation and task scheduling may significantly influence the efficiency and responsiveness of cloud-based applications, which becomes the investigation's core focus. The research proposes a novel approach known as DQNAC models, which use machine learning techniques to optimize resource allocation and task scheduling, enhancing the efficacy, flexibility, and performance of cloud computing. Significant gains over baseline methods are shown in the assessment of the suggested methodology utilizing a dataset of actual cloud workload traces. The research emphasizes the use of deep reinforcement learning-based approaches in optimizing cloud computing. It proposes them as a valuable strategy for enhancing the efficacy and sustainability of cloud-based applications.

D. Motivation

The elementary target of this investigation is to lead to the expanding body of knowledge in the area of cloud computing via the machine learning approaches to the issue of effectively distributing and organizing available resources and activities. Traditional methods have been employed to optimize resource allocation and task scheduling. Still, they don't always consider the unpredictable nature of cloud computing workloads, which may lead to inefficiencies that negatively affect application efficiency.

To overcome such constraints, the Deep Q-Networks and Actor-Critic (DQNAC) model is presented as a means of optimizing resource allocation and task planning via machine learning and deep neural networks [10-11]. Real-world cloud workload traces are used to test the proposed method, and the findings show that it has the potential to vastly improve resource utilization and operational efficiency compared to more conventional strategies. This study's ultimate goal is to integrate traditional wisdom with cutting-edge machine learning and deep learning methods for more efficient and productive cloud computing.

E. Objectives

The objectives of the research work are:

- To propose a new technique called Deep Q-Networks and Actor-Critic models that enhance cloud computing efficiency by optimizing resource allocation and task scheduling.
- To evaluate the proposed approach using a dataset of real-world cloud workload traces and compares it with traditional approaches.
- To show the efficacy of the suggested method in significantly improving resource utilization and overall performance in cloud computing.
- To bridge the gap between traditional approaches and modern machine and deep learning techniques for resource allocation and task scheduling in cloud computing.

F. Constitution of Research work

This research document is structured as follows: Section 2 summarizes the most important studies and their contributions to the field of resource allocation and task planning; Section 3 outlines the suggested approach in detail, including the datasets used and the computations involved in the DQNAC model; and Section 4 discusses the results of the performance evaluations and provides a brief synopsis. The study is then summed up with key findings and a strategy for future studies in Section 5.

II. RELATED WORK

Research difficulties associated with the evolving capacity of resources and their distribution in cloud computing have been well examined. In addition, many research documents and scholarly publications that contribute to the field have been reviewed. In this section, we provide an executive summary of each cited study.

Wang et al. [12] delivered an innovative method for assigning cloud platform resources: a fitness-oriented auction. Using this strategy, both the vendors and the customers are guaranteed to be a good "fit" in terms of their respective efficiency attributes. Next, they consider the method of distributing resources concerning monetary enhancement and system efficacy.

Another bio-inspired method for distributing work across VMs is the Cuckoo-optimization technique with particle swarm optimizer combo (Gawali & Shinde ; Gawali & Shinde)[13-14]. The efficiency of the task scheduler approach is improved by introducing a hybrid cuckoo-PSO, and the results are analyzed concerning cost as well as time.

Both an ML-oriented and an intuitive allocation technique were explored by Pahlevan et al. [15], and both were analyzed on the basis of service type, energy usage, network activity, and scalability across various data centers with the feature of mobility. Researchers employed a novel hyper-heuristic (HH) method for combining the best aspects of several ways and determined the most effective algorithm on the fly. To analyze efficiency, they proposed an integer parametric programming-based method for allocating VMs.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7179
***Article Received**: 21 April 2023 **Revised:** 09 June 2023 **Accepted:** 28 June 2023*

Tseng et al. [16] outlined the manner in which a genetic algorithm (GA) uses historical data to predict future resource requirements for an identified time window. In this study, a GA is used to predict available resources. In this paper, researchers offer a unique approach to VM allocation. Next, they addressed how to use the GA's predictions to determine how to allocate VMs to future time windows. Finally, the GA is used to find the optimal strategy for resource estimation in this simulation-based method.

Du et al. [17] took on the challenge of resolving the NP-hardness conundrum. The scenario proposes a sub-optimal solution that has the least intricacy, in which offloading decisions are made via sampling and semi-definite tradeoffs. The allocation of resources is accomplished using the concept of fractional operation with the support of the Lagrangian two-fold disintegration technique.

With the use of the novel algorithm, Gawali & Shinde, [14] developed a novel strategy whereby each task is completed before its first allotment to the resource pool. When allocating requests to available resources, the BAR+BATS (Bandwidth Aware Divisible Task Scheduling) optimization technique is employed to ensure that the workload and readily accessible connectivity of cloud computing assets are considered.

Ragmani et al. [18] considered the distribution of processing time. They deployed equipment and optimized the technique's efficacy through factoring variables, thereby rendering it more effective and assisting in regulating the workload in the cloud. Considering the processing time, optimum accomplishment duration, and load distribution, Velliangiri et al. [19] integrated a hybridized electric exploration with the evolutionary method. The research also wasn't concerned about VM load.

Cloud computing's efficacy has been boosted through ideal task planning due to the flexibility and cost of dynamically distributed resources [20]. The opposing rationale is paired with optimization methods to improve further the searching space's potential to be explored and exploited. To achieve an efficient task scheduler on a diverse CloudSim infrastructure at a low cost, researchers integrate Opposition-oriented training with Lion Optimization Approach (LOA) (Krishnadoss & Jacob,) [21] and developed hybrid Oppositional Lion Optimization Algorithm (OLOA).

The contrary lion optimization algorithm has been developed to maximize the task management approach's efficiency. The adversarial tactic is proposed to improve the optimization exploration objectives in reverse and forward perspectives. CloudSim is a technology that enhances the search capabilities of optimization, allowing for the ideal solution to be found Almalawi et al. [22]. Previous task-planning techniques, such as GA and PSO techniques, are contrasted to the lion optimization.

III. METHODOLOGY

A. Dataset

Anonymized workflow recordings across clustered VMs, as well as cloud systems, are available in the Workflow Trace Archive (WTA) (Workflow Trace Archive, n.d.; Versluis et al.,)[23], an archive for datasets. The WTA's goal is to standardize open-source information into a uniform format and provide a central repository for freely available workload datasets from which users can easily download the essentials.

Table I: Significance of WTA

Features	**Inclusions**
Domain	Engineering, Science, Industry
Platform	20
Workload	96
Workflow	48,310,465
Applications	Mixed
Task	1,900,898,384
User	1134
Group	76
Setup	67
Total hours	2,046,052,734
Objects (WTA format)	▪ Workload, ▪ Workflow, ▪ Task, ▪ Task_State Resource, ▪ Resource State, ▪ Data Transfer

With WTA, researchers and scientists have access to a massive database of workflow traces that accurately reflect the functioning of actual scientific processes/workflows [24]. These traces are utilized to build DRL models that can learn to optimize resource allocation and job scheduling for comparable processes in cloud computing. In addition, researchers may use these traces to test and refine their computational methods due to their openness.

The use of WTA is significant in this study because it allows access to actual-world information that can be used to train more precise and efficient machine learning methods. This, in turn, can lead to better resource allocation and task scheduling strategies that can improve the performance of cloud computing systems while reducing costs. Table 1 represents the significant features of WTA.

B. DQNAC

When applied to cloud computing, a Deep Q-Network (DQN) trained using actor-critic approaches could deliver a

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7179
***Article Received**: 21 April 2023 **Revised:** 09 June 2023 **Accepted:** 28 June 2023*

potent DRL model for optimizing resource allocation and job scheduling. The resultant model can gather knowledge over time, adjust to new circumstances, and prioritize work accordingly.

The DQN is used to approximate the optimum action-value function, and the actor-critic approach is applied to learn the policy/guidelines for choosing actions in the integrated DQNAC model. Two neural networks are used in the actor-critic approach: an actor-network, which makes action selections depending on the current state, and a critic-network, which provides feedback on the effectiveness of the actors' choices.

In this approach, the DQN is utilized to determine an approximation of the cumulative reward for a specific action in that state. The actor-network then uses the estimation to make a decision on what to do next. Finally, the actor-network receives input on the selected action from the critic-network, which it uses to fine-tune its policy periodically.

The DQNAC model can be trained using a combination of experience replay and policy gradient methods. Experience replay involves storing experiences in a replay buffer and randomly sampling batches of experiences to train the model. Policy gradient methods involve optimizing the policy parameters directly using gradient descent.

The resulting DRL model can be used to optimize resource allocation and task scheduling in cloud computing by allocating resources to different tasks based on the learned policy. The model can adapt to changing conditions, such as variations in workload or resource availability, and continuously learn and improve its policy over time.

C. Computations

The models' primary computation depends on three different modes: state, action and reward, which are elaborately defined and represented in the Table 2.

The combinatory process of DQNAC model to attain the objective of the research can be expressed as follows,

Let *S* be the set of states, *A* be the set of actions, $\mathcal{Z}$ be the reward function, and |D| be the discount factor. The goal is to find the optimal policy $\xi: S \rightarrow A$ that maximizes the expected cumulative reward.

Table II: Expression and Definition of State, Action, and Reward

Mode	Expression
State	*trace(S_t)* consisting of the current workflow, workload, and resource states, which is depicted as, $\boldsymbol{S_t} = \{\widehat{\boldsymbol{\alpha}}_{\boldsymbol{s}}, \widehat{\boldsymbol{\beta}}_{\boldsymbol{s}}, \widehat{\boldsymbol{R}}_{\boldsymbol{s}}\}$ (7) $\widehat{\boldsymbol{\alpha}}_{\boldsymbol{s}}$, a vector that represents the moment's workload, including active tasks, complete tasks, the mean duration of task execution, and resource requirements. $\widehat{\boldsymbol{\beta}}_{\boldsymbol{s}}$, a vector that contains information on the actual workflow status, such as the number of pending processes, their execution order, and their expected completion times. $\widehat{\boldsymbol{R}}_{\boldsymbol{s}}$, a vector that includes the actual resource status according to the specific platform scale, such as the number of readily accessible resources and the percentage of those currently in use.
Action	*trace(A_t)* consisting of the selected platform, setup, and task for execution, which is depicted as, $\boldsymbol{A_t} = \{\boldsymbol{I_\rho}, \boldsymbol{I_\varphi}, \boldsymbol{I_T}\}$ (8) I_ρ denotes the index of the selected platform, $I_\rho \in$ [1, 20] I_φ represents the index of the selected setup, $I_\varphi \in$ [1, 67] I_T indicates the index of the selected task, $I_T \in$ [1, 1900898384]
Reward	The reduction in time achieved by streamlining processes and decreasing workload, where: $\Delta \boldsymbol{R_t} = \left[\boldsymbol{H_t^{w_f+w_l}} - \boldsymbol{H_{t+1}^{w_f+w_l}}\right]$ (9) $H_t^{w_f+w_l}$,the total duration demanded to accomplish the workload and workflows at *t*. $H_{t+1}^{w_f+w_l}$,the total duration demanded to accomplish the workload and workflows at *t+1*, after performing the chosen *A*.

The DQN is used to approximate the optimal action-value function η(S,A), which is the expected cumulative reward of taking *A* in *S* and following the optimal policy thereafter. The DQN is trained using the Bellman equation, which relates the value of $\eta(S, A)$ to the value of $\eta(S',A')$ for the next state S' and action A':

$$\boldsymbol{\eta(S,A)} = \mathrm{e}\{(\mathcal{Z} + |\boldsymbol{D}|\boldsymbol{max}(\boldsymbol{\eta(S',A')})|(\boldsymbol{S},\boldsymbol{A}))\} \quad (10)$$

where e[..] denotes the anticipated value and $max(\eta(S',A'))$ is the maximum anticipated accumulative $\mathcal{Z}$ over all possible A' in the next S'. The actor-critic method consists of two neural networks: an actor network that selects actions based on the current state, and a critic network that evaluates the chosen *A*'s. Let $\xi(S,A;\beta)$ be the policy gradient model, parameterized by ß, which selects *A* in *S*, and let $f_v(S;\psi)$ be the state-value function parameterized by ψ, which evaluates the expected cumulative $\mathcal{Z}$ of being in *S* and following the policy ξ thereafter. The actor network is trained using policy gradient methods to maximize

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7179
***Article Received**: 21 April 2023 **Revised:** 09 June 2023 **Accepted:** 28 June 2023*

the expected cumulative reward, which can be expressed as follows:

$$\boldsymbol{P_G}(ß) = \mathrm{e}\{(\sum \boldsymbol{t} = \boldsymbol{0}_\infty |\boldsymbol{D}|_t \boldsymbol{Z_t} |\boldsymbol{S_0}, \boldsymbol{\eta})\} \quad (11)$$

where Z_t is the reward at time_step t and S_0 is the initial state. The critic network is trained using the temporal-variance ($\boldsymbol{\varpi}$) error, which measures the difference between the estimated value of the current state and the estimated value of the next state. The $\boldsymbol{\varpi}$ error can be expressed as follows:

$$\boldsymbol{\varpi} = \left[\boldsymbol{Z} + \left(|\boldsymbol{D}|\boldsymbol{v}(\boldsymbol{S'}; \boldsymbol{\lambda}) - \boldsymbol{v}(\boldsymbol{S}; \boldsymbol{\lambda})\right)\right] \quad (12)$$

where S' is the next state and $\boldsymbol{\lambda}$ are the parameters of the critic network.

Table III: Algorithm of DQNAC

Input: S, A, Z
Output: Convergence to the optimal policy, $\xi: S \rightarrow A$
1: Define S, A, Z
2: Define |D|
3: Approximate $\eta(S, A)$
$\eta(S, A) = \mathrm{e}\{\left(Z + |D| max\left(\eta(S', A')\right)\right|(S, A))\}$
4: Define $\xi(S,A;\beta)$ && $f_v(S;\psi)$
5: Train actor network using $\xi(S,A;\beta)$
$P_G(ß) = \mathrm{e}\{(\sum t = 0_\infty |D|_t Z_t |S_0, \eta)\}$ *//maximize the expected cumulative reward*
6: Train critic network using $\boldsymbol{\varpi}$
$\varpi = \left[Z + \left(|D|\upsilon(S'; \lambda) - \upsilon(S; \lambda)\right)\right]$
7: Train DQN;
Approximate η(S, A)
8: Perform Actor-Critic process
Learn → $\xi(S,A;\beta)$ && $f_v(S;\psi)$
9: Perform SAC-DAS
10: Repeat 3 to 9 until convergence

The DQNAC model combines these two methods by using the DQN to approximate the action-value function $\eta(S, A)$ and the actor-critic method to learn the policy $\xi(S, A; \beta)$ and the state-value function $\boldsymbol{\upsilon}(S;)$. The resulting model can be trained using a combination of experience replay and policy gradient methods to optimize resource allocation and task scheduling in cloud computing. Table 3 represents the step-wise algorithm of DQNAC model.

D. Experience Replay and Policy Gradient Methods

Experience replays and policy gradient methods are both popular reinforcement learning techniques. Experience replay involves storing experiences (i.e., state, action, reward, and next state) in a replay buffer and sampling batches of experiences randomly to update the Q-network. This helps to reduce the correlation between consecutive updates and stabilize the learning process. On the other hand, policy gradient methods aim to directly optimize the policy of the agent by estimating the gradient of the expected return with respect to the policy parameters. To integrate these techniques with the actor-critic model and deep Q-network, we can use a hybrid algorithm called the Stochastic Actor-Critic with Deterministic Action Selection (*SAC-DAS*) algorithm. SAC-DAS is an off-policy actor-critic algorithm that combines the benefits of both experience replay and policy gradient methods. Table 4 represents the procedural part of SAC-DAS.

Table IV: *Algorithm of SAC-DAS*

Input: S, A, Z
Output: Convergence to the optimal policy
1: Initialize critic network: c(S,A|θc) and the actor network: a(S|θa), Rb (replay buffer)
2: Repeat until convergence
2.1: Receive S;
2.2: Generate A_t;
$$A_t = a(S_t|\theta c) + \kappa_t$$
//κ_t denotes the noise operation included to encourage exploration.
2.3: compute A_t;
Observe: S_{t+1} and acquire Z_t
2.4: Store experience
$S_t, A_t, Z_t, S_{t+1} \rightarrow R_b$;
2.5: Sampling a random mini-batch from R_b
2.6: Compute target value;
$$\tau_i = Z_t + \upsilon c(S_{i+1}, c(S_{i+1}|\theta c)|\theta c')$$
2.7: update critic network
$$l = {}^1/_N \left(\sum \left(\tau_i - c(S_i, A_i|\theta c)\right)\right)^{2\nabla a\theta}$$
$$l = {}^1/_N \left(\sum \left(\tau_i - c(S_i, A_i|\theta c)\right)\right) \nabla a\theta \cdot c(S_i, A_i|\theta c)$$
$$\theta a = -(\Upsilon \nabla \theta c) l$$
//minimizing the mean squared error between the targeted and predicted value
2.8: Compute $\xi(S,A;\beta)$ using sampled experience
$$\nabla \theta a \sum |D| \approx {}^1/_N \sum \nabla A c(S, A|\theta c)|S = (S_i, A)$$
$$= a(S_i|\theta a) \nabla \theta a a$$

IV. PERFORMANCE ANALYSIS

Standard performance indicators can be used to evaluate task scheduling and allocation performance. Key performance measures and their outcomes in optimizing various algorithms using different scheduling approaches are discussed in this section. Some of the popular and relevant approaches like HH, GA, OLAO, BATS+BAR are considered for the evaluation of proposed DQNAC model with various performance metrics.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7179
***Article Received**: 21 April 2023 **Revised:** 09 June 2023 **Accepted:** 28 June 2023*

A. *Experimental Setup*

A computerized model of a real-time computational model of cloud infra (Calheiros et al.)[25] is used to evaluate the suggested DQNAC technique. Table 5 provides the VM and host configuration parameters, as well as generic data centre statistics, for the data centre utilized in the bespoke simulation scenario. The crucial characteristics of the experimental setup are summarized in the table below.

Table V: Experimental Setup

Component Characteristics		Value
Datacenter Configurations	Datacenter	2
	Host	2
	Storage (in Terabyte)	12 TB
	Execution Unit	8
	Task Execution Capacity (in Million Instructions Per Second	11600 mips
	Allocation Policy	SAC-DAS
	Pattern	X86
	Hypervisor	Xen
	Tracing gap	180
	Operating System (OS)	Linux
	RAM	80 GB
Host Configuration	Bandwidth	5 Gbps
	VM migration	Active
	RAM	40 GB
	Pattern	X86
	Hypervisor	Xen
	Tracing gap	180
	OS	Linux
VM Configurations	VMs	20
	Average RAM	1 GB
	Priority	1
	Allocation Policy	Dynamic Workload
	Snapshot and backup size	1000 bytes
	OS	Linux
	Script/Procedure element	1

B. *Outcome Evaluation and Discussions*

This subsection delivers a condensed explanation of the performance of the suggested DQNAC technique.

Response Time: Any strategy for allocating resources and organizing tasks ought to consider response time as a key performance indicator. Delay, or latency, refers to another name for response time (Raghunath & Rengarajan,)[26]. The total latencies result from combining the delays in data transmission and processing time (Dastjerdi et al.,) [27]. The delay in processing a task is known as the processing latency. In contrast, the transmission delay is the duration it takes for the required information to be sent among various resources. The following expression depicted in equation (1) can be used to determine the computational and transmission delays of any specific task:

$$\boldsymbol{L} = \left(\mathfrak{y}_{t_i} + \mathbb{T}_{t_i}\right) \quad (13)$$

where, L denotes the latency, $\mathfrak{y}_{t_i}$ indicates the communication delay of task t_i, and $\mathbb{T}_{t_i}$ denotes the transmission delay of task t_i.

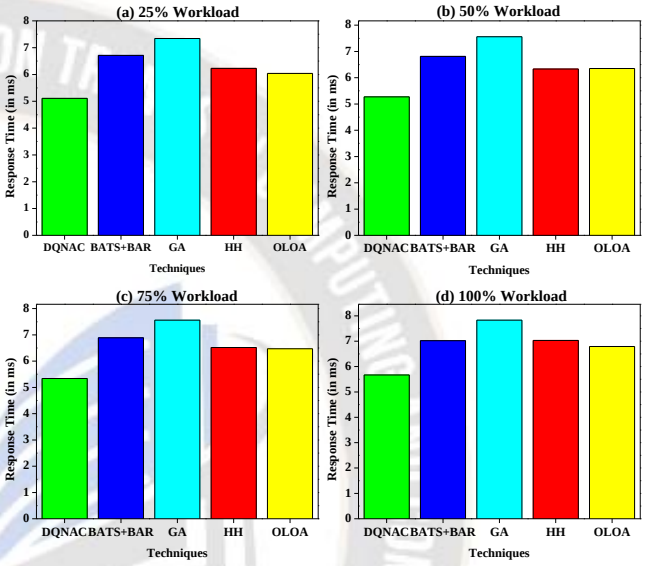

Figure 2: Analysis of Response Time

In cloud computing, the reaction time of various resource allocation and job scheduling approaches is shown in Figure 2. These methods are used for a variety of workloads. The importance of this indicator must be considered in the broader context of cloud computing since it directly correlates to customer satisfaction with the service they get.

The findings compare the reaction times of four distinct workload levels ((a) 25, (b) 50, (c) 75, and (d) 100 per cent) across five different resource allocation and task scheduling approaches (DQNAC, BATS+BAR, GA, HH, and OLOA). Each model's reaction time indicates how long it takes to complete a specific job at a given workload. The suggested DQNAC method's reaction time is lower than six at all workload levels, whereas the GA method's response time is higher. This indicates that the DQNAC approach to cloud computing resource allocation and job scheduling may be superior.

Processing Time: Average operational time refers to the time span required for the model/system to complete a scheduled task, ti. In this research, operational time excluded the waiting time spent for input/output and other overheads of the processes

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7179
***Article Received**: 21 April 2023 **Revised:** 09 June 2023 **Accepted:** 28 June 2023*

(Tanenbaum & Maarten van Steen,)[28]. Thus, the average operational time is expressed as,

$$ET = \left(F_{t_i} + I_{t_i}\right) \quad (14)$$

where, F_{t_i} denotes the completion time of the task, t_i and I_{t_i}represents the initial processing time of t_i.

Using four different proportional levels of workload (24, 48, 72, and 96), Figure 3 displays the average operation time (in seconds) for five distinct approaches to resource allocation and task scheduling. The findings reveal that when the workload level rises, the execution time rises along with it, showing that the system needs more time and resources to perform a similar task. DQNAC's execution time is consistently the quickest across all workloads, whereas GA's is consistently the slowest. The numbers show how various approaches to cloud computing's resource allocation and job scheduling stack up against one another. In cloud computing settings, DQNAC's shorter execution time implies that it could be a more efficient and effective way of allocating resources and scheduling tasks than other alternatives. Since GA takes longer to run, it's possible that it's not the best choice for these kinds of projects.

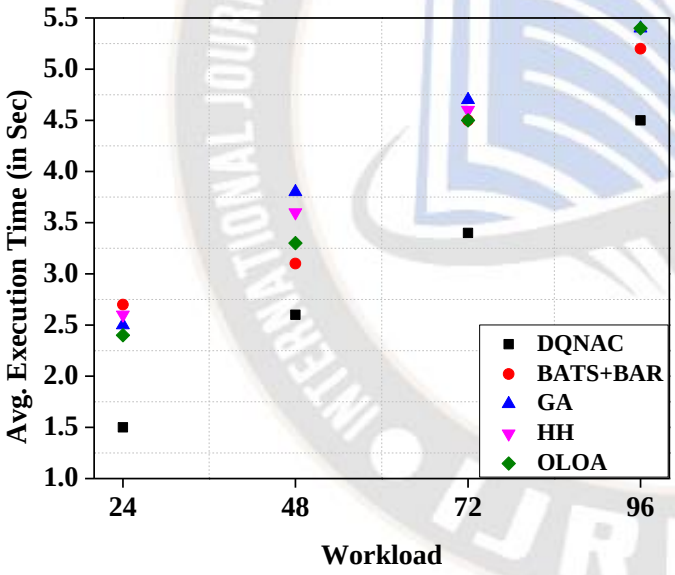

Figure 3: Analysis of avg. Operation Time

Makespan: The total amount of time needed to accomplish a workflow, is an important goal in task scheduling (Singh et al., 2017). When Makespan is kept to a minimum, programmes run quickly. Using Equation (15), Makespan can be determined as follows:

$$MS = \left(Y_{t_i} - X_{t_i}\right) \quad (15)$$

where, Y_{t_i} specifies the completion time of last executed task and X_{t_i} denotes the initial time of first executed task.

In Figure 4, we see how alternative cloud computing resource allocation and task scheduling strategies affect the average makespan (time required to complete all jobs in a workflow) in seconds.

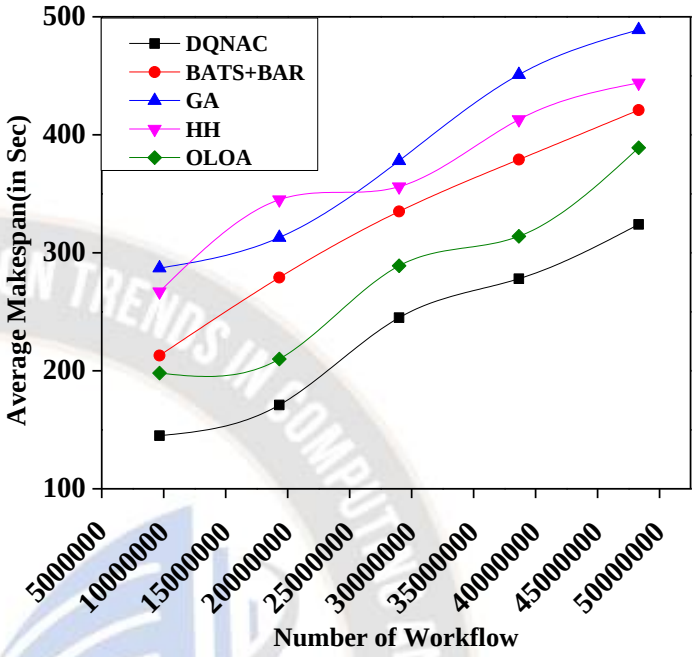

Figure 4: Analysis of Avg. Makespan

Statistical analysis reveals that DQNAC and OLOA techniques result in the shortest average makespan across all workflows. Furthermore, the average makespan for most processes is lower with BATS+BAR and GA than with HH, while it is more significant with DQNAC and OLOA.

When comparing several approaches to reducing the average makespan of workflows, DQNAC and OLOA are the most effective. However, it's vital to remain aware that the efficiency of these techniques might change based on the unique features and specifications of the actual workflows being carried out.

Bandwidth Utilization (BU): The model's effectiveness for allocating resources and scheduling tasks may be measured in terms of how much bandwidth is being utilized[29-30]. The formulation for evaluating the model with respect to this resource utilization is mainly focused on total amount data (*D*) transferred across the network, which is expressed as,

$$BU = \sum_{i=1}^{N}(D_{transmitted}) / 100 \quad (16)$$

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7179
***Article Received**: 21 April 2023 **Revised:** 09 June 2023 **Accepted:** 28 June 2023*

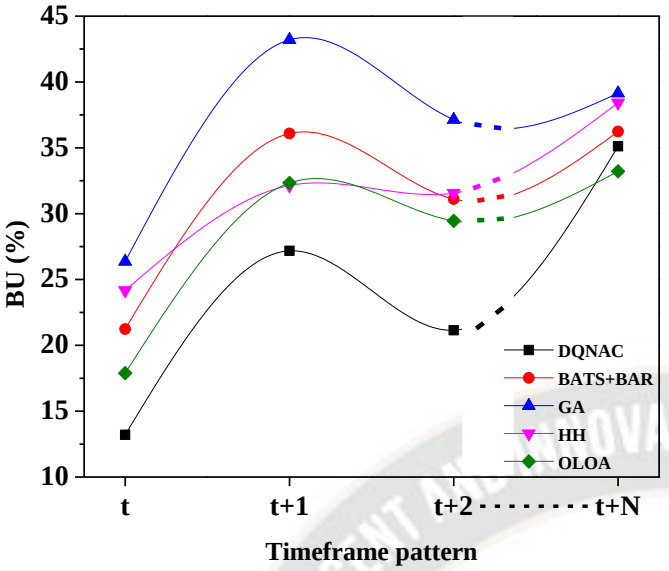

Figure 5. Analysis of Resource Utilization

Figure 5 shows the bandwidth utilization (BU) across a time period, indicated by the temporal pattern t, t+1, t+2, and t+N, for five distinct resource allocation and task scheduling approaches (DQNAC, BATS+BAR, GA, HH, and OLOA).

The percentage of the accessible bandwidth that is actually being utilized by a given platform or application is known as its bandwidth utilization. Bandwidth utilization is an essential metric in cloud computing that represents the efficacy of resource allocation and work scheduling techniques.

According to the observed data, bandwidth utilization shifts over time and differs across resource allocation and job scheduling techniques. DQNAC uses the most bandwidth among all temporal patterns, followed by HH and OLOA. Compared to the other period patterns, GA uses the least bandwidth, suggesting that it is not as efficient or effective as the others.

Compared to DQNAC and HH, BATS+BAR and OLOA exhibit reasonably stable bandwidth use over all temporal patterns. The data indicate that DQNAC is a strong contender for best practices in maximizing optimal bandwidth use in cloud computing settings.

Average Resource Utilization (ARU): In cloud infrastructure, the volume of CPU, memory, and input/output utilized in the VMs is measured by resource utilization. Equation (17) expresses the computation of determining ARU throughout operational time.

$$ARU = \sum_{i=1}^{m} (ET_{t_i})_{R_i} / MS \times T \qquad (17)$$

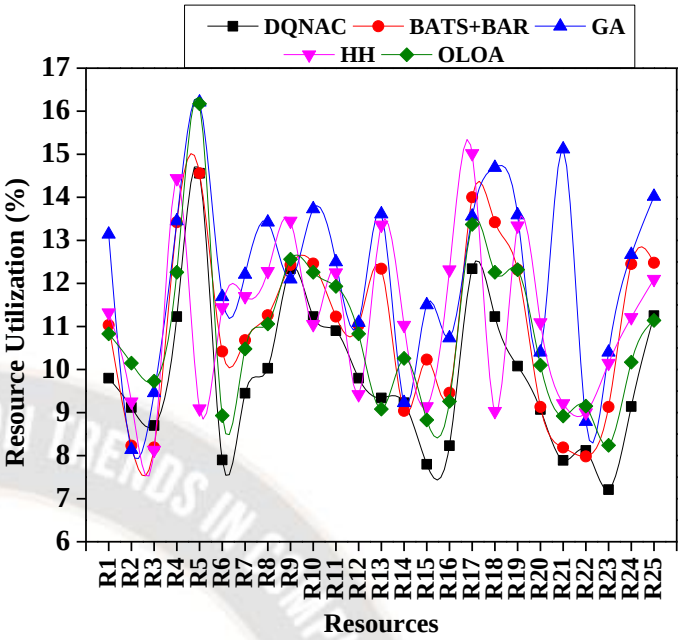

Figure 6: Analysis of Resource Utilization

The average utilization of each resource (R1 through R25) scheduled using DQNAC, BATS+BAR, GA, HH, and OLOA is shown in Figure 6. From the result, we can deduce that various scheduling techniques provide varying degrees of resource utilization. Some scheduling strategies may work better than others for a given resource in some contexts. The ARU for the DQNAC approach is 9.8396, the lowest of any method. This implies that time and efforts are not wasted because of inefficient resource allocation. Higher than DQNAC, the ARU for the BATS+BAR technique is 11.0044. The results imply that this approach could be more practical than DQNAC, but it does a respectable job of balancing workloads and allocating resources.

Compared to DQNAC and BATS+BAR, the ARU for the GA technique is 12.218, which is greater. This implies that this method is not desirable in terms of time or energy efficiency compared to the other two approaches. The HH technique has an ARU of 11.1928, which is quite similar to the average results of the BATS+BAR approach. The ARU for the OLOA technique is 10.8116, which is a bit lower than the BATS+BAR method's average statistics. This data demonstrates that this approach also excels at scheduling and allocating resources for various projects. Overall, DQNAC seems to be the most effective approach in terms of both resource management and the assignment of tasks.

V. CONCLUSIONS AND FUTURE ENHANCEMENT

For optimal resource allocation and task planning in cloud computing, the study suggests a novel way of utilizing machine and deep learning techniques termed Deep Q-

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7179
***Article Received**: 21 April 2023 **Revised:** 09 June 2023 **Accepted:** 28 June 2023*

Networks and Actor-Critic (DQNAC) models. Using traces from actual cloud workloads, the study assesses the DQNAC method. It shows that it greatly outperforms comparable versions of BATS+BAR, GA, HH, and OLOA regarding resource utilization, average makespan, execution time, and response time. The results indicate that DRL-based methodologies could potentially be used as a powerful and effective tool for optimizing cloud computing, improving the efficiency and flexibility of cloud-based applications.

For future improvements, we are interested in investigating the manner in which Generative Adversarial Networks (GANs), Recurrent Neural Networks (RNNs), and other advanced deep learning methods can be applied to better optimize cloud computing's resource allocation and task planning. In addition to improving the efficacy and flexibility of cloud-based systems, these methods have also shown promise in other sectors. For example, DRL-based approaches in the cloud may benefit from using real-time data analytics and predictive modeling. Cloud systems can better distribute resources and optimize task scheduling in real-time by continually monitoring and adjusting to evolving workloads and the availability of resources.

REFERENCES

[1] Jula, A., Sundararajan, E., & Othman, Z. (2014). Cloud computing service composition: A systematic literature review. Expert Systems with Applications, 41(8), 3809–3824. https://doi.org/10.1016/j.eswa.2013.12.017

[2] Kumari., V.Y.R., Madhav., T. B., & Kumar., L. R. (2015). Efficient and Secure Scheme for Distributed Data Storage Systems. International Journal of Computer Science and Information Technologies, Vol. 6 (1) , 2015, 839-843

[3] Khan, T., Tian, W., Zhou, G., Ilager, S., Gong, M., & Buyya, R. (2022). Machine learning (ML)-centric resource management in cloud computing: A review and future directions. Journal of Network and Computer Applications, 204, 103405. https://doi.org/10.1016/j.jnca.2022.103405

[4] Kumar, Y., Kaul, S., & Hu, Y.-C. (2022). Machine learning for energy-resource allocation, workflow scheduling and live migration in cloud computing: State-of-the-art survey. Sustainable Computing: Informatics and Systems, 36, 100780. https://doi.org/10.1016/j.suscom.2022.100780

[5] Khan, A.I., Alsolami, F., Alqurashi, F., Abushark, Y.B. and Sarker, I.H., 2022. Novel energy management scheme in IoT enabled smart irrigation system using optimized intelligence methods. Engineering Applications of Artificial Intelligence, 114, p.104996.

[6] Ding, D., Fan, X., Zhao, Y., Kang, K., Yin, Q., & Zeng, J. (2020). Q-learning based dynamic task scheduling for energy-efficient cloud computing. Future Generation Computer Systems, 108, 361–371. https://doi.org/10.1016/j.future.2020.02.018

[7] Wang, B., Liu, F., & Lin, W. (2021). Energy-efficient VM scheduling based on deep reinforcement learning. Future Generation Computer Systems, 125, 616–628. https://doi.org/10.1016/j.future.2021.07.023

[8] Saraswathi, A. T., Kalaashri, Y. R. A., & Padmavathi, S. (2015). Dynamic Resource Allocation Scheme in Cloud Computing. Procedia Computer Science, 47, 30–36. https://doi.org/10.1016/j.procs.2015.03.180

[9] Aron, R., & Abraham, A. (2022). Resource scheduling methods for cloud computing environment: The role of meta-heuristics and artificial intelligence. Engineering Applications of Artificial Intelligence, 116, 105345. https://doi.org/10.1016/j.engappai.2022.105345

[10] Zhou, G., Wen, R., Tian, W., & Rajkumar Buyya. (2022). Deep reinforcement learning-based algorithms selectors for the resource scheduling in hierarchical Cloud computing. Journal of Network and Computer Applications, 208, 103520–103520. https://doi.org/10.1016/j.jnca.2022.103520

[11] Raghunath B. H., & Aravind H. S. (2023). An Efficient FPGA-Based Dynamic Partial Reconfigurable Implementation. International Journal of Intelligent Systems and Applications in Engineering, 11(1s), 183–192. Retrieved from https://ijisae.org/index.php/IJISAE/article/view/2471

[12] Sarker, I.H., Khan, A.I., Abushark, Y.B. and Alsolami, F., 2022. Internet of things (iot) security intelligence: a comprehensive overview, machine learning solutions and research directions. Mobile Networks and Applications, pp.1-17. https://doi.org/10.1007/s11036-022-01937-3

[13] Wang, H., Kang, Z., & Wang, L. (2016). Performance-Aware Cloud Resource Allocation via Fitness-Enabled Auction. IEEE Transactions on Parallel and Distributed Systems, 27(4), 1160–1173. https://doi.org/10.1109/tpds.2015.2426188

[14] Gawali, M. B., & Shinde, S. K. (2017). Standard Deviation Based Modified Cuckoo Optimization Algorithm for Task Scheduling to Efficient Resource Allocation in Cloud Computing. Journal of Advances in Information Technology, 8(4), 210–218. https://doi.org/10.12720/jait.8.4.210-218

[15] Gawali, M. B., & Shinde, S. K. (2018). Task scheduling and resource allocation in cloud computing using a heuristic approach. Journal of Cloud Computing, 7(1). https://doi.org/10.1186/s13677-018-0105-8

[16] Andrew Hernandez, Stephen Wright, Yosef Ben-David, Rodrigo Costa, David Botha. Optimizing Resource Allocation using Machine Learning in Decision Science. Kuwait Journal of Machine Learning, 2(3). Retrieved from http://kuwaitjournals.com/index.php/kjml/article/view/195

[17] Pahlevan, A., Qu, X., Zapater, M., & Atienza, D. (2018). 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. https://doi.org/10.1109/tcad.2017.2760517

[18] Tseng, F.-H., Wang, X., Chou, L.-D., Chao, H.-C., & Victor. (2018). Dynamic Resource Prediction and Allocation for Cloud Data Center Using the Multiobjective Genetic

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 8s
DOI: https://doi.org/10.17762/ijritcc.v11i8s.7179
***Article Received**: 21 April 2023 **Revised:** 09 June 2023 **Accepted:** 28 June 2023*

Algorithm. IEEE Systems Journal, 12(2), 1688–1699. https://doi.org/10.1109/jsyst.2017.2722476

[19] Du, J., F. Richard Yu, Chu, X., Li, W.-Y., & Lu, G. (2019). Computation Offloading and Resource Allocation in Vehicular Networks Based on Dual-Side Cost Minimization. IEEE Transactions on Vehicular Technology, 68(2), 1079–1092. https://doi.org/10.1109/tvt.2018.2883156

[20] Ragmani, A., Elomri, A., Abghour, N., Moussaid, K., & Rida, M. (2019). FACO: a hybrid fuzzy ant colony optimization algorithm for virtual machine scheduling in high-performance cloud computing. Journal of Ambient Intelligence and Humanized Computing, 11(10), 3975–3987. https://doi.org/10.1007/s12652-019-01631-5

[21] Rajesh Patel, Natural Language Processing for Fake News Detection and Fact-Checking , Machine Learning Applications Conference Proceedings, Vol 3 2023.

[22] Velliangiri, S., Karthikeyan, P., Arul Xavier, V. M., & Baswaraj, D. (2021). Hybrid electro search with genetic algorithm for task scheduling in cloud computing. Ain Shams Engineering Journal, 12(1), 631–639. https://doi.org/10.1016/j.asej.2020.07.003

[23] Khan, A. I., Alghamdi, A. S. A., Abushark, Y. B., Alsolami, F., Almalawi, A., & Ali, A. M. (2022). Recycling waste classification using emperor penguin optimizer with deep learning model for bioenergy production. Chemosphere, 307, 136044.

[24] Krishnadoss, P., & Jacob, P. (2019). OLOA: Based Task Scheduling in Heterogeneous Clouds. International Journal of Intelligent Engineering and Systems, 12(1), 114–122. https://doi.org/10.22266/ijies2019.0228.12

[25] Almalawi, A., Khan, A.I., Alsolami, F., Abushark, Y.B. and Alfakeeh, A.S., 2023. Managing Security of Healthcare Data for a Modern Healthcare System. Sensors, 23(7), p.3612. https://doi.org/10.3390/s23073612

[26] Versluis, L., Mathá, R., Talluri, S., Hegeman, T., Prodan, R., Deelman, E., & Iosup, A. (2020). The workflow trace archive: Open-access data from public and private computing infrastructures. IEEE Transactions on Parallel and Distributed Systems, 31(9), 2170-2184. https://doi.org/10.1109/tpds.2020.2984821

[27] Abushark, Y. B., Khan, A. I., Alsolami, F., Almalawi, A., Alam, M. M., Agrawal, A., ... & Khan, R. A. (2022). Cyber security analysis and evaluation for intrusion detection systems. Comput. Mater. Contin, 72, 1765-1783.

[28] Calheiros, R. N., Ranjan, R., Beloglazov, A., De Rose, C. A. F., & Buyya, R. (2010). CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience, 41(1), 23–50. https://doi.org/10.1002/spe.995

[29] Raghunath, K. M. K., & Rengarajan, N. (2018). Evolving Optimal Response Time and Synchronized Communication on Integrating Fuzzy Logic Using Infrared Sensor and Sound Detecting Sensor in WSN. Sensor Letters, 16(8), 606–613. https://doi.org/10.1166/sl.2018.3993

[30] Dastjerdi, A. V., Gupta, H., Calheiros, R. N., Ghosh, S. K., & Buyya, R. (2016). Fog Computing: principles, architectures, and applications. Internet of Things, 61–75. https://doi.org/10.1016/b978-0-12-805395-9.00004-6

[31] Tanenbaum, A. S., & Maarten van Steen. (2007). Distributed Systems. Prentice Hall.

[32] Singh, P., Dutta, M., & Aggarwal, N. (2017). A review of task scheduling based on meta-heuristics approach in cloud computing. Knowledge and Information Systems, 52(1), 1–51. https://doi.org/10.1007/s10115-017-1044-2

[33] Workflow Trace Archive. (n.d.). wta.atlarge-Research.com. Retrieved April 24, 2023, from https://wta.atlarge-research.com