International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
IJISAE
ISSN:2147-6799 www.ijisae.org Original Research Paper

A Cloud-Based Adaptive Multi-Agent Deep Deterministic Policy Gradient Technique-Based Hybrid Optimisation Algorithm for Efficient Virtual Machine Migration and Task Scheduling

Gurpreet Singh Panesar[1], Dr. Raman Chada[2]

Submitted: 25/08/2023 **Revised**: 13/10/2023 **Accepted**: 26/10/2023

Abstract: TIn cloud computing settings, virtual machine (VM) migration and task scheduling are essential components that are meant to improve system efficiency and resource use. In this work, we present a unique method for effective task scheduling and virtual machine migration: the Cloud-Based Adaptive Multi-Agent Deep Deterministic Policy Gradient (AMS-DDPG). To improve resource allocation in cloud settings, the AMS-DDPG strategy combines the advantages of Adaptive Multi-Agent and Deep Deterministic Policy Gradient (DDPG). In order to further improve the AMS-DDPG technique's performance, we provide an iterative idea of War and Rat Swarm (ICWRS). This idea does parameter optimization inside the AMS-DDPG approach using War Strategy Optimization (WSO) and Rat Swarm Optimizer (RSO). In order to effectively adjust the settings of the AMS-DDPG approach and enhance its convergence and performance, WSO and RSO simulate war and swarm tactics, respectively. Through the use of simulated cloud computing scenarios, different workload patterns and system configurations are taken into consideration while evaluating the suggested technique. Our test findings show that, in comparison to conventional techniques, the AMS-DDPG methodology with the ICWRS approach performs better. By optimizing VM migration and task scheduling, the AMS-DDPG approach increases overall system efficiency, lowers energy consumption, and improves resource usage. Furthermore, by presenting a thorough hybrid optimization approach that makes use of deep reinforcement learning and optimization strategies inspired by nature, our study adds to the rapidly expanding area of cloud computing. The possibility of merging numerous advanced approaches for tackling complicated resource allocation challenges in cloud systems is shown by the combination of DDPG, AMS-DDPG, ICWRS, WSO, and RSO. Future research aiming at using intelligent optimization techniques to improve the scalability and efficiency of cloud computing systems has a potential direction thanks to this study.

Keywords: *Deep Deterministic Policy Gradient (DDPG), Iterative Concept of War and Rat Swarm (ICWRS), Rat Swarm Optimizer (RSO), Multi-Agent Deep Learning, Adaptive Multi-Agent System, War Strategy Optimization (WSO).*

1. Introduction

Blockchain, cloud computing, and artificial intelligence are just a few of the new, wireless technologies that have garnered the Internet of Things (IoT)[1] many benefits. Since cloud computing can guarantee a wide range of features, it has attracted significant interest from academics, government agencies, and businesses alike . Fog computing, utility computing, and so on are all built right in. The demand model also strives for maximum efficiency in terms of both resources and output[2]. Therefore, several types of clouds, such as cloud federation, mobile, hybrid, private, and public, have been deployed to cater to the wide range of needs. Software as a Service (SaaS), Infrastructure as a Service (IaaS), and Platform as a Service (PaaS) are the three main categories of cloud services . With the PaaS, developers have access to a wide range of resources for enhancing cloud services, including virtual and physical machines, programming languages, operating systems, and control structure design patterns. The software as a service model provided a platform for accessing cloud-based software, allowing customers to hire developers and use the software on a pay-as-you-go basis. In addition, the IaaS provides access to virtualized physical assets, computers, and storage. In addition, the availability of resources may be increased or decreased in response to fluctuations in demand.

The cloud data centre has been used by service providers all over the globe to deliver cloud services. Furthermore, critical hardware is considered the backbone of cloud services[3]. The high cost of maintaining cloud data centres due to their excessive energy usage is highlighted here. As a result of a number of factors, including inefficient data centre cooling, low server utilisation, and underutilised network hardware, data centre energy has been largely ignored. The cloud computing method is based on virtualization, which has created an isolated environment for a wide variety of applications. Functionalities such as hardware resource abstraction, streamlined access, and dynamic resource management have also been made available. In addition to improving a system's adaptability, it has simplified the process of replicating virtual instances and spreading service clients for isolation among nodes. Server virtualization has emerged as an important method

[1]*Research Scholar, Chandigarh University, Punjab, India*
[2]*Professor, Chandigarh University, Punjab, India*
** Corresponding Author Email: Gurpreet.e9842@cumail.in*

for maximising cloud computing by allowing several servers to use a single, widely distributed data center's resources. Therefore, the most popular methods for reducing power consumption and optimising resource utilisation inside a virtualized data centre are discussed.

Significant approach to migrate the VM via the source into the target host was verified due to important constraints such as specific destination hosts[4]. Using these methods, migration time may be cut in half by making better use of available network capacity. In addition, virtual machine (VM) migration to the cloud has made use of the pre-copy paradigm. In this context, the term "data rate" refers to the filthy rate at which the VM's memory state has been modified during a migration. Edge clouds are challenging in comparison to traditional cloud data centres because of issues like live VM migration, short migration times, and high quality of service[5]. This is because the wide area network used by edge clouds is far more bandwidth-constrained than the model used by data centres. In recent years, many meta-heuristic algorithm models have been developed to address the pressing problem of VM consolidation. Multi-objective optimisation problems as resource waste, migration time, resource utilisation, energy usage, and migration overhead have been taken into account by major VM consolidation approaches. To address these issues, we develop a novel virtual machine (VM) migration method using cloud-based model-based reinforcement learning.

To construct a novel, highly effective cloud-based VM migration and job scheduling framework using federated learning and a hybrid algorithm model for cross-database or cross-host propagation.Second, using a mixed heuristic approach, create an AMS-DDPG model for relocating virtual machines and assigning work to them in the cloud. Construct a new model for an algorithm called ICWRS, which incorporates the two current algorithms. It is used to fine-tune the model's operating settings for optimal efficiency. Achieve the multi-objective functions by optimising for CPU use, energy consumption, makespan, migration cost, active servers, and quality of service. To examine the suggested framework from a variety of experimental perspectives, with the goal of achieving the guaranteed performance[6].

This work has been divided into the following parts. In Tier II, we survey the existing literature, in Tier III, we propose a model of VM migration and task scheduling in the cloud network using a hybrid optimisation strategy, in Tier IV, we use a hybrid heuristic algorithm based on the iterative concept of war and rat swarm to optimise the model's parameters, in Tier V, we implement an adaptive multi-agent DDPG and objective function, and in Tier VII, we present our findings and draw our conclusions.

2. Literature Survey

Cloud computing has witnessed unprecedented growth, reshaping the landscape of modern computing by providing scalable and cost-effective solutions for businesses and individuals alike. This paradigm-shifting technology offers on-demand access to a vast pool of computing resources through the internet, enabling users to deploy and scale their applications and services without the need for extensive hardware investments. Virtualization, a core component of cloud computing, abstracts physical hardware into virtual resources, allowing multiple virtual machines (VMs) to run on a single physical machine (PM). This efficient resource utilization leads to significant cost savings and improved operational efficiency for cloud service providers. However, along with its numerous advantages, cloud computing poses several challenges, and energy consumption within cloud data centers (DCs) stands out as a critical concern. As the demand for cloud services continues to surge, the power consumption of servers, networking equipment, cooling systems, and other infrastructure components within DCs has escalated, resulting in significant operational costs and environmental impact. Cloud service providers are increasingly striving to strike a balance between delivering high-quality services and minimizing energy usage to improve cost-effectiveness and environmental sustainability. In the research by Khan et al. (2018), they developed a Genetic Algorithm-based approach to optimize VM placement in a multi-cloud environment. By considering various Quality of Service (QoS) constraints, their study aimed to intelligently distribute VMs across multiple clouds, thereby improving cloud resource utilization[7].

In a study conducted by Ko et al. (2019) they introduced Particle Swarm Optimization (PSO) to improve the energy efficiency of VM placement, in cloud systems. Their research focused on selecting machines (PMs) for VM allocation with the goal of minimizing energy consumption while maintaining application performance. This approach proved effective in enhancing energy efficiency within cloud data centers [8].

Shao et al. (2017) explored the use of the Ant Colony System (ACS) for VM allocation in cloud data centers. Their objective was to optimize resource utilization and energy efficiency by mimicking foraging behavior. They aimed to achieve a distribution of VMs across PMs thereby improving resource allocation [9].

On the hand Zhang et al. (2020) proposed an approach called Q learning based VM allocation for placement of VMs in cloud data centers. This approach aimed to optimize resource utilization while meeting Quality of Service (QoS) requirements with a focus on adaptability and real time decision making, in changing cloud environments [10].

In their study Reji and Selvakumar (2019) employed the Deep Deterministic Policy Gradient (DDPG) approach to streamline the consolidation and migration of machines, in cloud data centers. Their technique yielded allocation of VMs resulting in utilization of resources and energy efficiency. This demonstrates the capability of DDPG to effectively handle action spaces [11].

In their work Wang and colleagues (2021) introduced an approach using DDPG to allocate virtual machines (VMs) in multi cloud environments. Their strategy showed performance compared to methods particularly in terms of energy efficiency and resource utilization. This study highlights the potential of DDPG, in addressing VM allocation challenges, within cloud settings [12].

Mnih and colleagues (2015), as Lillicrap and colleagues (2016) have made significant contributions to the application of DDPG in complex optimization problems, such as VM allocation. Their research is highly influential in the field of reinforcement learning, for control tasks. Their work paved the way for future innovations in reinforcement learning-based approaches[13].

Sutton and Barto (2018) delved into the theoretical foundations of reinforcement learning, offering an in-depth introduction to the field. This fundamental knowledge is essential for understanding and applying RL algorithms, including DDPG, in various optimization tasks[14].

Proximal policy optimisation techniques were described by Schulman et al. (2017) and are pertinent to work on RLbased VM allocation. Aligning with the goal of learning optimum VM allocation policies, their study advanced our knowledge of optimising policies in reinforcement learning[15].

3. ADVANCED VM Migration and Task Scheduling Model for Cloud Networks: A Hybrid Strategy for Optimisation

3.1. Cloud System Overview

All users have been able to use virtualized resources that are scalable and available on demand thanks to the cloud. In this case, the user only has to pay for the resources that they really utilize since they are available under the pay-per-use model. Additionally, by reducing their administration responsibilities, users may spread the pool of computing resources that have been allocated. Furthermore, under- and over-provisioning of resources was disregarded, and the expense of using hardware—which has been seen as a source of incentive for businesses moving their operations into the clouds—was kept to a minimum. IaaS, PaaS, and SaaS are among the many methods that have been portrayed as being over the cloud and specifically tailored for cloud computing. Additionally, based on the requirements, the user has chosen a service and an employment strategy.

In this case, the cloud has been used at different levels across domains, resulting in a variety of actors having access to personal data stored on the cloud.

Data owner: It is thought to be the primary player. Therefore, the decision has been made as to whether or not to use cloud-hosted services or host data there [16]. The ownership of the data might include several parties, particularly if the data was co-produced by multiple parties or is going to be aggregated in terms of multiple parties.

Cloud Service Provider (CSP): As the entity that hosts the data, CSP is regarded as the main player in this scenario. Cloud-specific processing such as segmentation, data mining, statistics, indexation, research, storage, transfer, duplication, and more may be carried out over private data by it. As a result, it is represented by both software entities and persons. Additionally, it has taken covert steps that jeopardized the data's privacy. The CSP has emphasized the gain over the consumer behavior data, which has led to the definition of the data as a significant source of financial benefit. As a result, the CSP has made use of the data to create statistics that will improve services and create new enterprises.

Cloud Service Broker (CSB): In this case, the actor is the business that has totaled the values of traditional cloud services based on the users of such services [17]. It has taken on three significant responsibilities including integration, aggregate broking, and customisation. The only function available to the CSB in the limited circumstances is consultation.

Cloud-based service: The actor has portrayed a range of cloud-based apps and programs that are used to carry out services such as document management, medical applications, accountancy, cloud storage, collaboration tools, and so forth. The majority of the time, the application has obtained access to personal information in order to provide some kind of therapy. Furthermore, it has guaranteed that its services are safe and haven't compromised user privacy.

The administrator of cloud-based services is regarded as the service owner. Control and improvement of the services are the main responsibilities [18]. When the administrator has been changed through CSP, the administrator has obtained the same incentive as the CSP to provide user information.

Third parties providing cloud-based services are considered to be the ones who have used or are connected to the use of cloud-based services. Although this actor has generally been seen to be trustworthy, in the worst situations it has been thought to be untrustworthy and may jeopardize data privacy [19]. In Fig.1, this mechanism is shown.

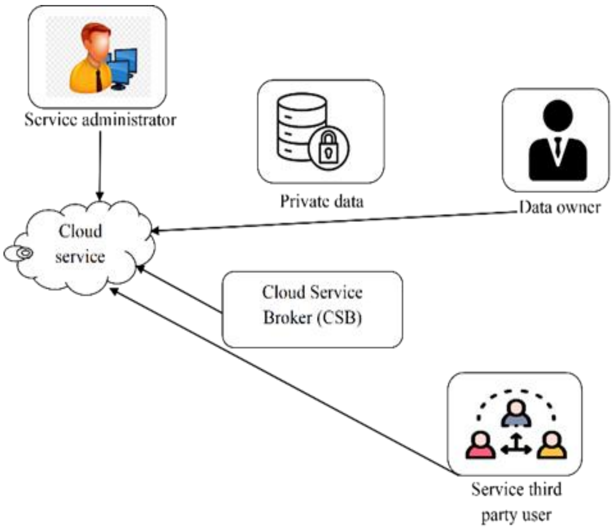

Fig.1 Depiction for the overview of the cloud system

3.2. VM Migration and Task Scheduling Issues

Cloud computing relies heavily on virtual machines (VMs), and users often request completed tasks from the cloud. Moreover, users have submitted requests along with varied resource needs to the cloud system, and these requests have been handled through the task queue [20]. In addition, a VM manager has sent the input jobs from the task queue to the task schedulers. The available VMs have been assigned to the jobs by the task schedulers. work scheduling aims to reduce the overall completion time of all input tasks by first detecting the estimated resources so that the specific work may be scheduled on time, then making more efficient use of those resources, and then scheduling the tasks to be completed. There are two main stages where management of cloud resources is carried out. Initially, it was the task scheduling that established the value of the virtual machine in carrying out the tasks [21]. Task scheduling has been done for a variety of purposes, including but not limited to power management, load balancing, increased resource usage, and reduced execution time. In addition, the second stage involves the allocation of virtual machines (VMs) to actual hardware.

Virtual machines can't be successfully deployed on hosts because they need resources that aren't being provided by the hosts. Other problems have arisen when assigning VMs to hosts due to insufficient host resources compared to the VMs' acquired resources.

• Internet consumers now have access to virtualized software programs, computer power, and storage [22]. In addition, QoS has made use of the service level agreements made available through cloud services in order to make best use of the readily available resources. Finding that the server was overloaded led to the conclusion that there was a problem with the virtual machine migration procedure.

Furthermore, memory size and range have been genuinely dynamic, making it challenging to transmit propagating bandwidth. Through the use of memory self-ballooning, efficient pre-copy termination, write throttling, memory compression, and de-duplication, the VM migration strategy has successfully reduced the VM size.

It has also run into a wide variety of challenges, such as low bandwidth, unpredictability in the network, longer delays, and a higher packet loss percentage.

Memory and storage may both be moved using VM migration strategies thanks to the shared network connection [23]. As an added bonus, postponing the VM migration's completion may sometimes result in a significant gain in terms of time.

The difficulty rate for finding suitable stopping sites has been optimized. Limitations in application throughput, a high packet loss rate, and a sluggish response time are all indicators of migration noise, which have been reduced thanks to the VM migration method. Furthermore, the challenge that has given unanticipated behavior has deteriorated the predictive applications' preemptive resource needs.

Why Data transmitted through the wrong network lines during VM migration, making security a concern; this is particularly problematic at greater communication distances. In addition, the limited isolation afforded by shared resources has allowed malicious VMs to access other VM address spaces and engage in malicious activity. • The data integrity has been guaranteed during the VM migration process by making full use of complex cryptographic functions [24].

3.3. Explanation of the Planned Approach

Virtualized resources are those that can be made available to users through the internet and include things like computing power, storage space, and software programs. In addition, the Quality of Services has made use of varying degrees of service agreements and has been delivered through cloud service providers in order to make the most efficient use of readily available resources. Optimal VM allocation has been used to reduce data center energy usage. Consolidating virtual machines has improved the efficiency of the cloud data center's energy consumption when the off-duty servers are taken into account [25]. The virtual machine (VM) has been deployed in a way that maximizes efficiency while minimizing expenses. Figure 2 presents a new model built in the cloud using an adaptive and heuristic algorithm that takes into account the shortcomings of the classic model.

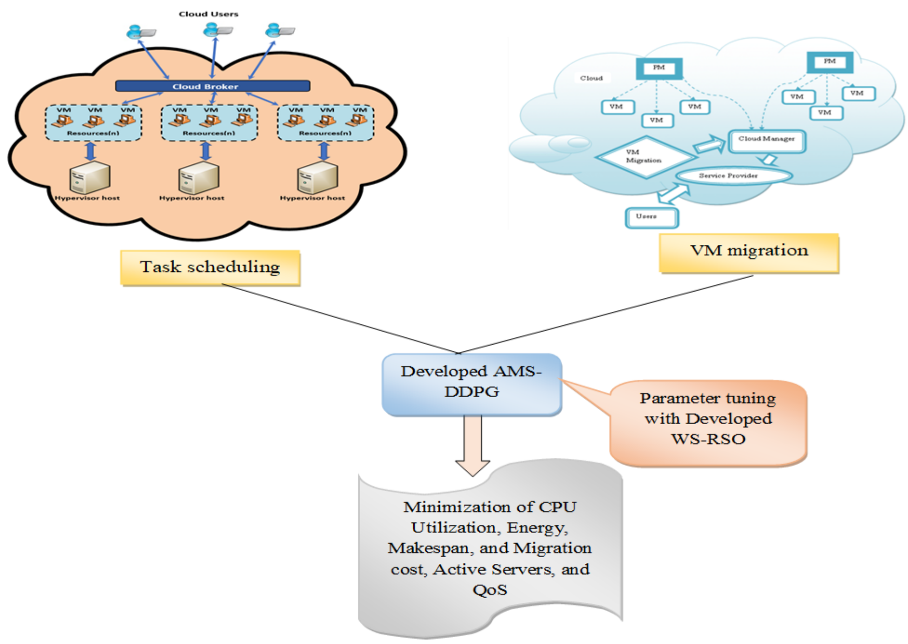

Fig.2 New paradigm shown for virtual machine migration and scheduling of tasks

A novel paradigm for virtual machine migration and work scheduling has been developed to address various issues with the status quo. In addition, a novel AMS-DDPG method is created to accomplish cloud-based job allocation and virtual machine migration using a hybrid heuristic approach. To further address the shortcomings of the standard hybrid model, a new algorithm model known as ICWRS has been developed. Therefore, the AMS-DDPG technique's performance has been optimized by the use of this algorithm model. Thus, the suggested approach has achieved several multi-objective functions during the task scheduling phase, including CPU usage, energy, makespan, migration cost, active servers, and quality of service [26]. The proposed architecture has therefore been shown to reliably provide better cloud performance and task scheduling rates across a wide range of experimental validations.

4. Hybrid Heuristic Algorithm for Parameter Optimization: An Iterative Concept of War and Rat Swarm

4.1. Current WSO

Based on their fitness rating, all 26 warriors have a likelihood of becoming either a commander or a king. Furthermore, the other troops have benefited from the commander and king's dissemination around the battlefield.

Method of attack: Two different approaches have been taken into consideration. The soldier adjusts the position based on the kind's location and the commander [27]. Every soldier had the same rank at the start of the conflict. In this instance, the soldier carried out the plan, leading to a promotion. Every soldier's weights and rankings have been adjusted based on the success approach. As the fight draws to a close, the troops stay quite close to their objective, as does the army commander and the monarch. Equation (1) expresses it.

$$A_z(a+1) = A_z(a) + 2 \times \omega \times (A - B) + RD \times \left(C_z \times B - A_z(a)\right) \tag{1}$$

The weight is shown here as, the king and commander's location is indicated as, and the new position is designated as.

Rank and weight updating: Every seeking agent's location update has been dependent on the interaction between each soldier's rank, Commander, and King [28]. Furthermore, each soldier's rank is determined by their record of achievement in combat. Additionally, each soldier's rank indicates their proximity to one another, which is taken into account while determining their fitness level. Compared to the traditional approach, the weighted factors changed linearly, while the weight changed exponentially with the factor of.

When a soldier's fitness in a new place is reduced to that of their former position, that previous location is considered acquired.

$$A_z(a+1) = \left(A_z(a+1)\right) \times \left(FT_x \geq FT_{ps}\right) + \left(A_z(a)\right) \times \left(FT_x < FT_{ps}\right) \tag{2}$$

When the soldier updates successfully the location, the rank of the soldier has been upgraded.

$$RK_y = (RK_y + 1) \times (FT_x \geq FT_{ps}) + (RK_y) \times (FT_x < FT_{ps}) \quad (3)$$

The new weight factor is defined in Eq. (4) as a function of rank.

$$C_z = C_z \times \left(1 - \frac{RK_y}{max\_i\,te}\right)^{\xi} \quad (4)$$

Defense plan: In this case, the second plan's location update has been based on the sites. Additionally, there has been no change in the ranking or weight for updating.

$$A_z(a+1) = A_z(a) + 2 \times \omega \times (A - A_{RD}(a)) + RD \times C_z(cm - A_z(a)) \quad (5)$$

Because the prior approach contained the position of the random soldier, the examination of the war strategy in this case demonstrated maximum seeking space while assimilating. The soldier has completed additional stages and updated its positions in light of the greater significance of. The soldier has completed fewer stages and revised its places in light of the lower value.

Replacement/relocation of weak soldiers: For each iteration, it has detected the weak soldier that has minimized fitness. Here, the multiple replacement techniques have been tested. Here, the simplest techniques have been replaced through the weak soldier along with the random soldier and it is given in Eq. (6).

$$A_w(a+1) = lb + RD \times (ub - lb) \quad (6)$$

Further, the second technique has been relocating the weak soldier closer to the median of the whole army on the war field and it is given in Eq. (7). Thus, this technique has enhanced the convergence rate of the behaviour of the algorithm.

$$A_w(a+1) = -(1 - RDn) \times (A_w(a) - md(A)) + A \quad (7)$$

4.2. Current RSO

In general, the RSO algorithm model has included both the chasing as well as the attacking behaviour and thus it has aided to design of the algorithm model. The behavior of the rat has been aggressive in some cases, in which it has resulted in the death of some animals.

Chasing prey: The rats are normally social animals that chase their food [29]. To determine the behavior of the rat that has regarded the better searching solutions that have the knowledge of the location of prey. Here, the other updated location in accordance with the better location solution has been obtained so far. It is derived using Eq. (8).

$$\vec{Z} = Y \bullet \vec{Z}_e(f) + X \bullet \vec{Z}_g(f) - \vec{Z}_e(f) \quad (8)$$

Here, the position of the rat and the improved ideal solution are shown as and. Equations (9) and (10) have been used to derive additional and parameters.

$$Y = V - f \times \left(\frac{V}{mx_{it}}\right) \quad (9)$$

Here, $f = 0,1,2,\cdots,mx_{it}$

$$X = 2.RD \quad (10)$$

Moreover, the terms V and X are used to define the random number. Then, the random parameters Y and X are more responsible for performing the better exploitation and exploration phase in terms of iterations. The fighting process of the rats along with their prey along with the time of hunting has been mathematically derived in Eq. (11).

$$\vec{Z}_e(f+1) = \left|\vec{Z}_g(f)\right| - \vec{Z} \quad (11)$$

The rate's next updated location is shown here as. Additionally, by using a better location, it has updated the locations of other seeking agents and preserved the original location [30]. Through the process of modifying the parameter values, both exploration and exploitation have been accomplished. Consequently, the best solution for certain operators has been automatically stored by the proposed algorithm model.

4.3. Suggested ICWRS

The cloud's virtual machine migration and task scheduling infrastructure has been significantly enhanced with the creation of the new hybrid algorithmic mode, or ICWRS. The two traditional algorithms have improvements and drawbacks. Better trade-offs between the exploitation and exploration stages have been achieved using the WSO model [31]. However, when combined with other meta-heuristic models, there aren't any multi-objective functions. It is this method that has shown the multimodal and unimodal functions. It has provided an improved fix for optimization problems. Nevertheless, it has not been able to address the problems of objective optimization or multi-objective optimization. As a result, several limitations have been addressed using novel formulations.

The WSO model is used to carry out the update when the value is divisible by 5 and 7; otherwise, the RSO model is used.

Thus, its pseudo-code is given in Algorithm 3.

Algorithm 3: ICWRS		
The army size, dimension of issues, and ranks of every soldier and certain parameters are initialized		
The fitness value of every searching agent is validated		
The best search agent is determined		
	If t is divisible by 5 and 7	
		Update the position using WSO
	Else	
		Update the position using RSO
	End if	
End		
Return better solution		

Then, Fig. 3 shows the flowchart model.

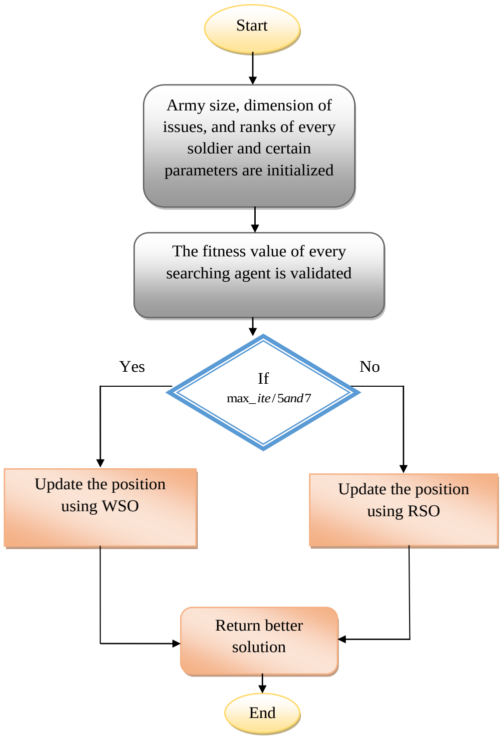

Fig.3 The ICWRS model's flowchart

5. Vacuum Migration and Task Scheduling in the Cloud with Objective Function and Adaptive Multi-Agent Ddpg

5.1. Deep Deterministic Policy Gradient

The initialized configuration has been provided as the technique's input in this step.

Taking into account the DDPG -dependent framework, the architecture is the same because the agents have both the actor networking model and the critic incorporated. When the actor is taken into account, it is seen as the policy function, in which it has interacted to monitor the state, considers continuous action via deterministic policy, and achieves the immediate reward. As a result, the critic updated the settings by using the action value functions. The agent's ultimate goal is to increase the long-term return by identifying the best resource allocation strategy.

State space: In this case, the system's current state is determined by monitoring the utility rates of each base station, which is supported by both the Edge Cloud (EC) and the Back-end Cloud (BC), as well as the current task that has to be scheduled through BC or EC.

Action Space: After taking into account each state, the agents have determined the bandwidth and virtual machine resources that will be used to carry out the task. In addition, the includes three parameters. In this case, the bandwidth is considered as, the CPU cycle is represented as, and the cloud has a value of either 0 or 1, indicating that the job has been assigned to the appropriate resources.

Policy Gradient (PG) algorithms are a class of reinforcement learning algorithms used for training policies in a reinforcement learning setting. These algorithms directly learn a parameterized policy that maps states to actions, aiming to maximize the expected cumulative reward. The policy is typically parameterized by some learnable parameters denoted as θ.

The objective function to maximize in policy gradient methods is the expected cumulative reward, also known as the expected return [32]. For a policy $\pi\theta$, the anticipated return, represented by $J(\mu)$, is as follows:

$E\tau\sim\pi\theta[R(\tau)] = J(\theta)$

Wherever

The trajectory $\tau=(s0,a0,r1,s1,a1,r2,\ldots)$ is produced by the policy $\pi\theta$.

The cumulative reward earned in trajectory τ is denoted by $R(\tau)$.

Finding the ideal policy parameters θ^* that maximize $J(\theta)$ is the aim. Gradient ascent is used to update the policy parameters in a manner that raises the anticipated return in order to accomplish this. The policy gradient, represented by the symbol $\nabla J(\theta)$, is the gradient of the anticipated return with respect to the policy parameters [33].

The formula for the policy gradient is: $\nabla J(\theta)=E\tau\sim\pi\theta[\nabla\theta\log\pi\theta(\tau)\cdot R(\tau)]$

Wherever

The gradient of the trajectory's log probability under the policy $\pi\theta$, relative to the policy parameters θ, is denoted by $\log\pi(\tau)$.

In order to maximize the anticipated return, the policy parameters are then changed in the direction of the policy gradient:

$$J(\theta t) = \theta t+1 + \alpha \quad (12)$$

where learning rate (α) is expressed.

In actual application, environmental samples are utilized to estimate the policy gradient. To stabilize and enhance the training process, a number of algorithms are used, including REINFORCE, PPO (Proximal Policy Optimization), A3C (Asynchronous Advantage Actor-Critic), and TRPO (Trust Region Policy Optimization).

5.2. Deterministic Policy Gradient (DPG) Algorithms:

Deterministic Policy Gradient (DPG) algorithms are a class of reinforcement learning algorithms that focus on learning deterministic policies in the context of continuous action spaces. In contrast to stochastic policies that output a probability distribution over actions, deterministic policies directly map states to specific actions.

The objective of DPG is to maximize the expected cumulative reward for a deterministic policy. Let's define the deterministic policy as μθ(s), where θ are the learnable parameters of the policy. The expected return for a deterministic policy μθ is denoted as J(θ) and is given by:

$$J(\theta)=Es0\sim\rho\mu,at\sim\mu\theta[\sum t=0\infty\gamma trt] \quad (13)$$

*s*0 is the initial state sampled from the state distribution *ρμ*.

at=*μθ*(*st*) is the action chosen by the deterministic policy at time *t*.

rt is the immediate reward received at time *t*.
γ is the discount factor.

The deterministic policy gradient is the gradient of the expected return with respect to the policy parameters θ. It is denoted as ∇J(θ) and can be expressed as:

$$\nabla(\theta)=Est\sim\rho\mu[\nabla\theta\mu\theta(st)\cdot\nabla aQ\mu(st,a)|a=\mu\theta(st)] \quad (14)$$

where:

Qμ(*s*,*a*) is the action-value function for policy *μ*, representing the expected cumulative reward starting from state *s*, taking action *a*, and following policy *μ* thereafter.

In practice, the policy parameters are updated using gradient ascent to maximize J(θ):

$$\theta t+1=\theta t+\alpha\nabla J(\theta t)$$

where α is the learning rate.

DPG algorithms often use an actor-critic architecture, where the actor is the deterministic policy μθ(s) and the critic approximates the action-value function Qμ(s,a). The critic helps in estimating the policy gradient by providing the value function gradient ∇aQμ(s,a).

Deep DPG (DDPG): This DPG variation is recognized for having both the policy that has been approximated and the critic together with the deep neural network model. In this case, the off-policy algorithm and sample trajectories via the experiences' replay buffer—which has been restored after the training phase—have been specified as the DDPG. It may also make use of the intended network.

Fully connected networks (FCNs) have been used primarily as actor-networks and critic networks in the DDPG-dependent architecture. As a result, they have large trainable weights and are able to achieve the global discriminative characteristics of the task sequences.

5.3. MA-DDPG that adapts

Adaptive Multi-Actor Deep Deterministic Policy Gradient (Adaptive MA-DDPG) is a reinforcement learning algorithm designed for cooperative multi-agent scenarios. It extends the DDPG algorithm to handle environments with multiple interacting agents. Each agent maintains its own deterministic policy, mapping states to continuous actions using actor and critic networks.

The innovation in Adaptive MA-DDPG lies in its adaptable communication strategy among agents [34]. Agents communicate by sending and receiving messages to coordinate actions. The level of communication is dynamically adjusted based on performance. If agents are performing well individually, communication is reduced to streamline computations. Conversely, when coordination is crucial for improved performance, communication is heightened to facilitate effective collaboration.

During training, agents receive their local observations and messages from other agents, using this information to update their policies. The critic helps estimate the action-value function, enabling computation of the policy gradient for actor updates. The actors aim to maximize expected cumulative rewards for each agent based on this gradient.

By dynamically regulating communication, Adaptive MA-DDPG allows agents to strike a balance between collaboration and autonomy, crucial in achieving efficient and effective policies in complex cooperative tasks. This adaptability enhances scalability and robustness in multi-agent environments.

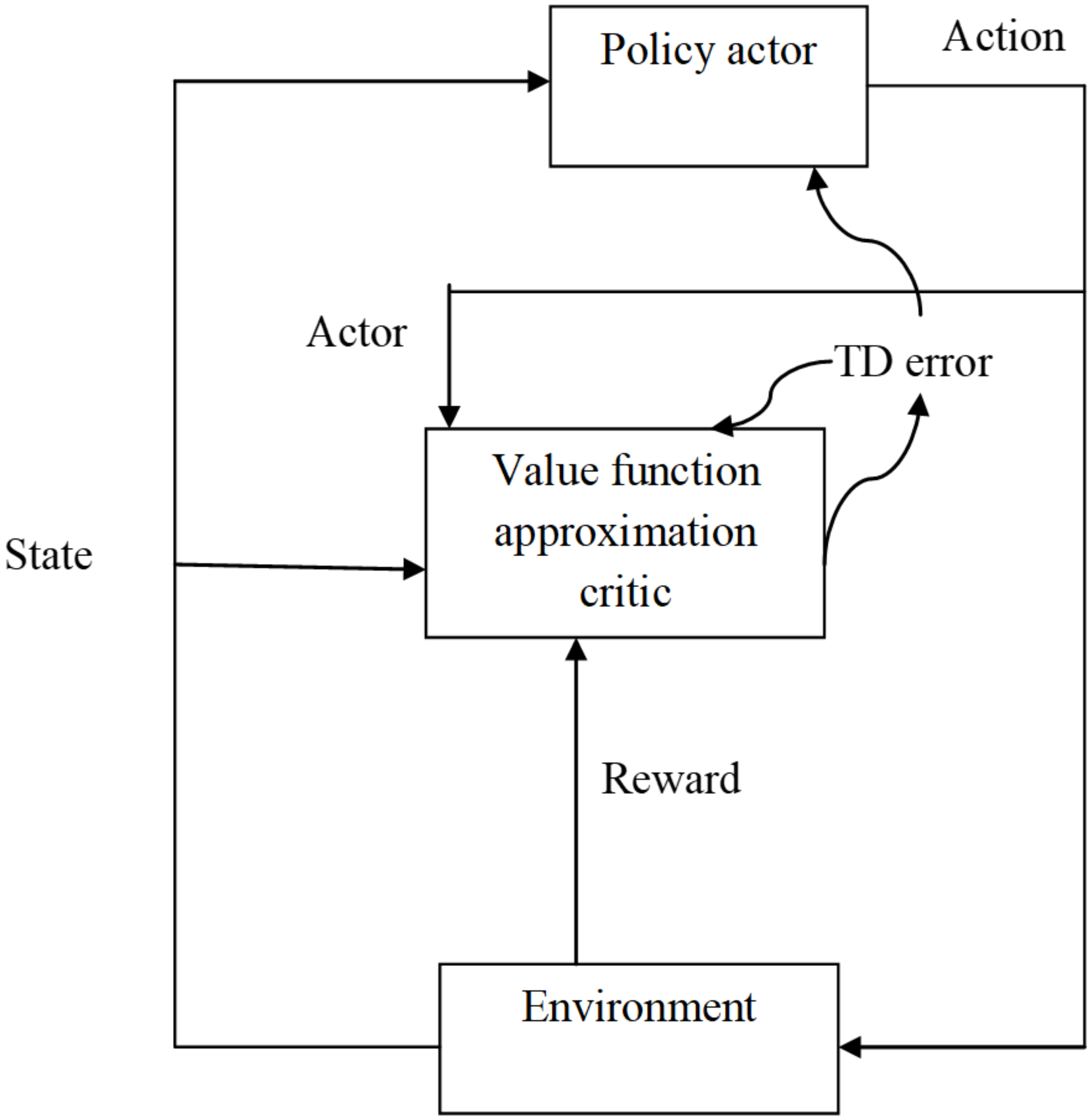

Fig.4 Depiction of MA-DDPG model with parameter tuning

5.4. Definition of the Objective Function and Constraints

Here, a full explanation of the restrictions employed in the provided framework for performance improvement is provided.

Quality of Service (QoS) may be described as the "objective, system-related characteristics that offer insight into the network/transmission level performance of the delivery service.

" It is stated in Equation (15).

$$ZZ_{vv} = \frac{\sum_{xy=1}^{4} ZZ_{xy} WT_{xy}}{\sum_{xy=1}^{4} WT_{xy}} \quad (15)$$

Here, the QoS value, denotes the weight of the criterion, is the index of the assigned weight, and is the the QoS value of the criterion.

Energy Use: Equation (16) expresses it.

$$ec_{cc} = \int_{ti1}^{ti2} cp_{cc}\, dti \quad (16)$$

Here,

the energy consumption is termed as

the power consumption is depicted as .

CPU utilization: Eq. (17) provides the derivation.

$$cp_{cc} = \left(cp_{cc}^{max_{cc}^{min}} * UR_{cp} + cp_{cc}^{idl} \right) \quad (17)$$

In this case, the CPU usage rate is shown as UR_{cp}, $cp_{cc}^{\max}$ and $cp_{cc}^{\min}$ is the power usage at its highest and lowest points.

Migration Cost: This includes both downtime and the actual migration duration. It is stated in Equation(18).

$$cs_{com} = cs_{req} + cs_{rep} + cs_{path} \quad (18)$$

The route establishment cost is shown here as cs_{path} the cost of the reply message appears as cs_{rep}, the request message's cost is referred to as cs_{req} and the price of the overhead related to communication is stated as cs_{com}. Fig. 5 shows this mechanism in action.

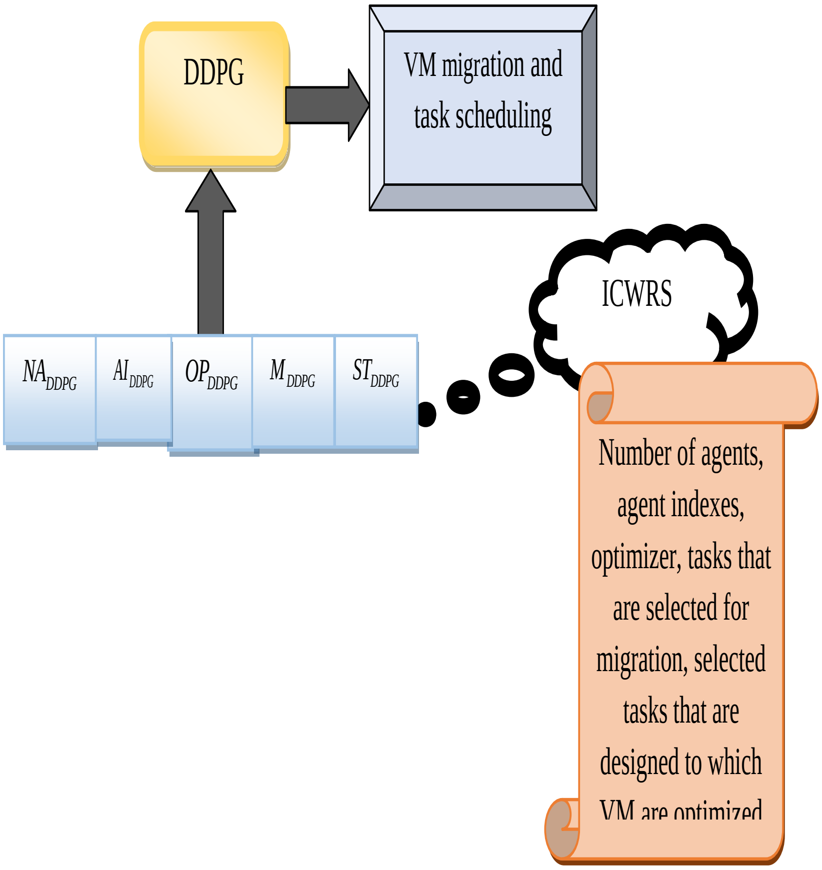

Fig.5 Solution encoding representation

6. Results and Discussion

6.1. Simulation Setup

The suggested model for task scheduling and migration was coded up in MATLAB 2020a, and then an analysis was carried out on the results. In order to do a comparison, we used the programs Egret Swarm Optimization (ESO)-AMS-DDPG and Dingo Optimizer (DO)-AMS-DDPG.

6.1.1. Configuration setup

The configuration settings for the recently developed scheduling model are provided in Table II. Within this model, it has been made easier to carry out the scheduling process by modifying the configuration values in a number of different ways.

TABLE I. CONFIGURATION VALUES FOR THE NEWLY PROPOSED SCHEDULING MODEL

Configuration Number	***No of Vm***	***No of Task***
1	10	100
2	20	200
3	30	300
4	50	500

6.2. Determination over the cost function in the interest of optimization

It appears that in the context of task scheduling and migration models using diverse algorithms, you've compared the cost function values obtained from different models, specifically ICWRS-AMS-DDPG, ESO-AMS-DDPG, DO-AMS-DDPG, WSO-AMS-DDPG, and RSO-

AMS-DDPG. The comparison was visualized in Fig. 6, demonstrating that the ICWRS-AMS-DDPG model achieved lower cost function values compared to the other models.

6.2.1. Cost Function

The cost function, often denoted as (J(θ)) or simply (f(x)), is a mathematical representation of the objective you're trying to optimize in your task scheduling and migration model [35]. It quantifies the 'cost' associated with a particular configuration, policy, or set of decisions in your model.

6.2.2. Model Comparison

Comparing the values of the cost function across models can give us insights, into how they perform relative to each other. In this case it is clear that the ICWRS AMS DDPG model outperformed the models mentioned as it achieved lower cost function values.

This difference in performance highlights the importance of the ICWRS AMS model. Suggests that it is a more efficient and effective approach for solving the task scheduling and migration problem compared to the other models considered.

This comparison plays a role in demonstrating both the effectiveness and efficiency of the proposed ICWRS AMS model providing strong evidence for its significance, in addressing task scheduling and migration challenges.

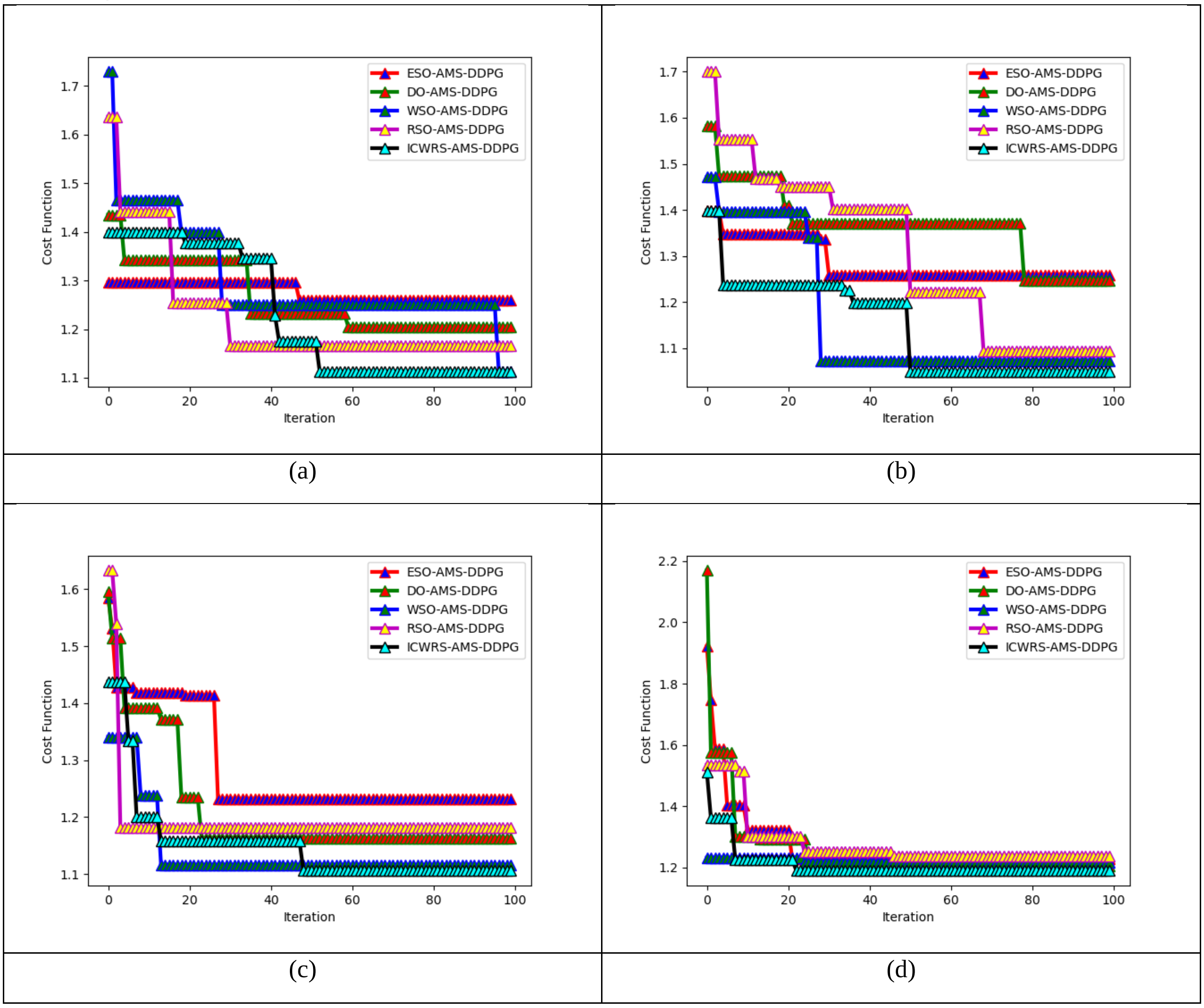

Fig.6 shows the algorithmic validation of the recommended cloud-based task scheduling and virtual machine migration framework using the cost function in the following configurations: a) Configuration 1, b) Configuration 2, c) Configuration 3, d) Configuration 4.

6.3. Validation for the purpose of scheduling tasks and migration

Fig. 7 shows the validation of the newly proposed task scheduling and migration model in terms of classifiers and algorithms by adjusting the Active sensors, CPU usage, energy consumption, Make span, migration cost, and QoS. The recommended ICWRS-AMS-DDPG model has shown superior value than other models when taking into account the value of active sensor and QoS. As a result, the value of CPU usage, energy consumption, make span, and migration cost have all increased. Therefore, compared to alternative ESO-AMS-DDPG, DO-AMS-DDPG, WSO-AMS-DDPG, and RSO-AMS-DDPG models, the recommended ICWRS-AMS-DDPG model offers superior values of 2%, 25%, 3%, and 3% for the value of an active sensor. As a result, the classifier comparison reveals that in contrast to DRL+DQN, EPBLA, EEHVMC, and AMS-DDPG, the suggested ICWRS-AMS-DDPG has shown a lower value of active sensors at 5%, 10%, 20%, and 50%.

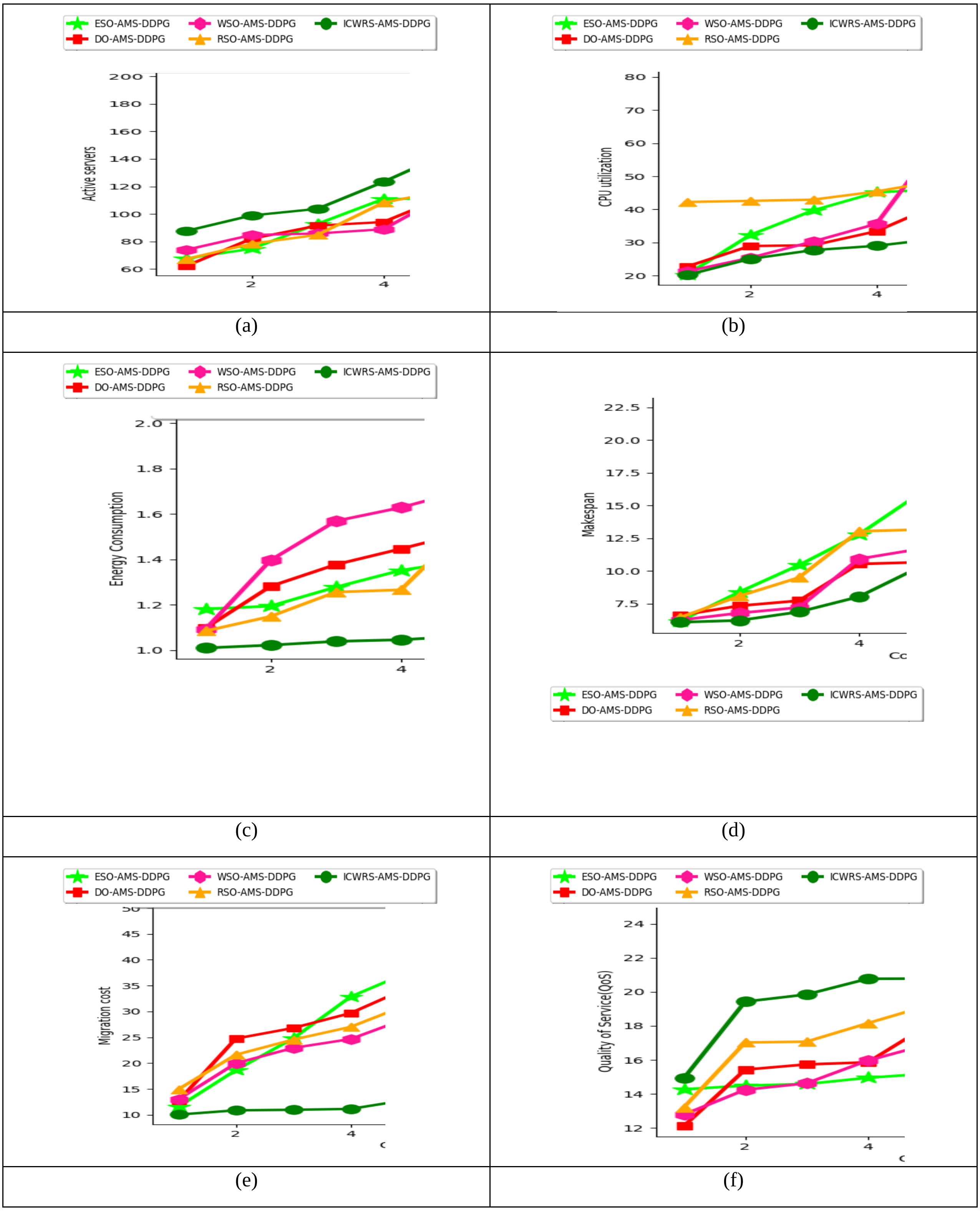

Fig.7 Validation in terms of algorithms for the process of tsak scheduling and migration in the proposed model by means of a) Active sensors, b) CPU utilization, c) Energy consumption, d) Makespan, e) Migration cost and f) QoS

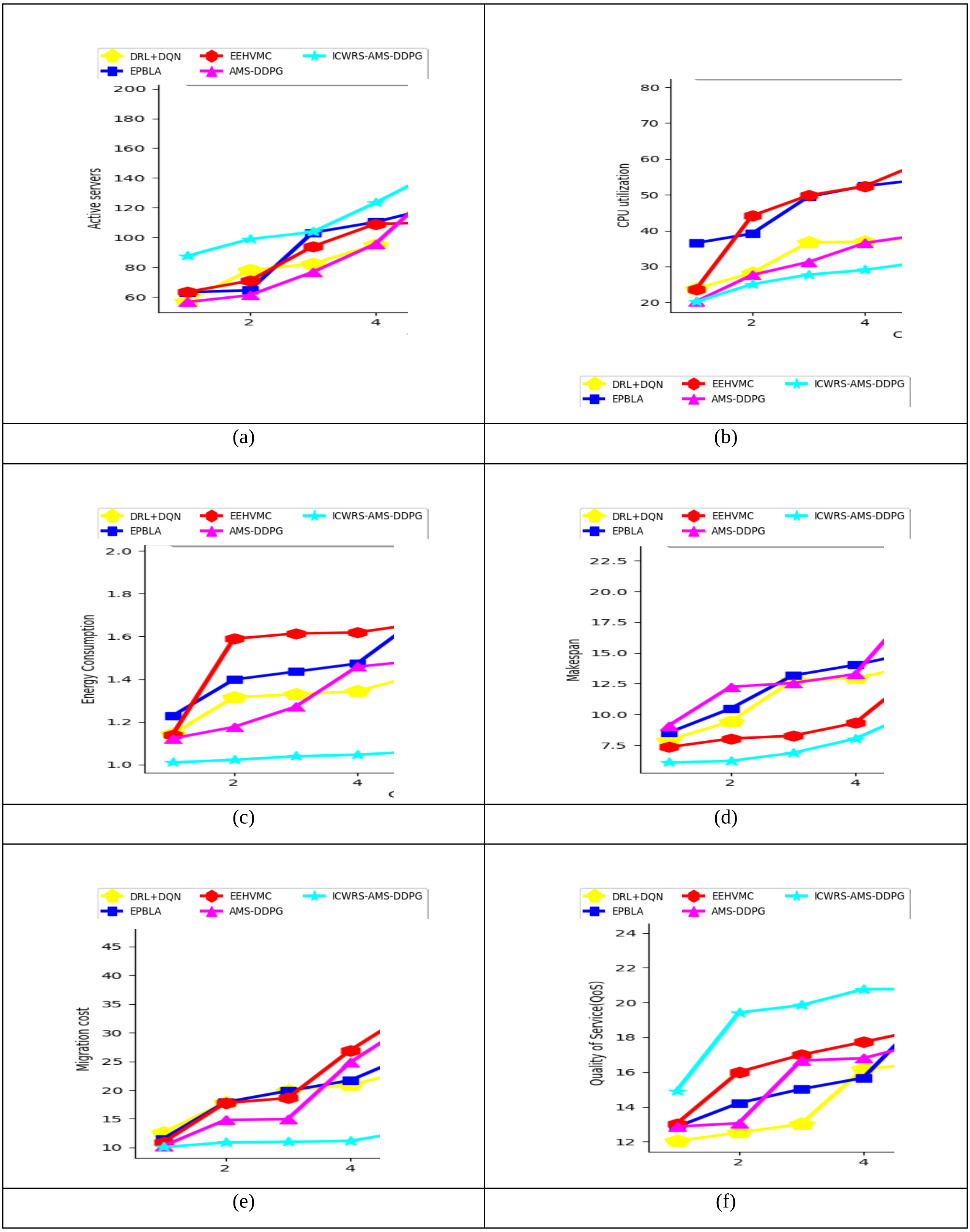

Fig 8. Validation in terms of classifiers for the process of tsak scheduling and migration in the proposed model by means of a) Active sensors, b) CPU utilization, c) Energy consumption, d) Makespan, e) Migration cost and f) QoS

7. Conclusion

This study presents novel research that develops a novel paradigm to solve many issues with conventional virtual machine (VM) migration and work scheduling techniques. With a particular emphasis on virtual machine migration and task scheduling, this study aims to improve the efficacy and efficiency of cloud-based operations by using cutting-edge computational tools such as MATLAB 2020a and innovative algorithmic techniques.

Improving Task Assignment and Virtual Machine Migration Using an Innovative Hybrid Heuristic Method

The main objective of this study revolves around the creation of a method known as Adaptive Multi Agent Deep Deterministic Policy Gradient (AMS DDPG). The purpose of this technique is to tackle the difficulties related to shifting machine tasks and distributing workloads in cloud environments. In order to streamline these procedures and effectively assign tasks the AMS DDPG approach incorporates a heuristic algorithm that combines elements from the Deep Deterministic Policy Gradient (DDPG) mechanism. The goal of the AMS-DDPG methodology is to improve overall system performance in the cloud by optimizing resource allocation.

Taking Care of Limitations with the ICWRS Algorithm Model

Additionally, a novel iteration formulation was created to overcome the limitations of the normal hybrid model, resulting in the development of the Iterative Concept of War and Rat Swarm (ICWRS) algorithm model. This model contributes to the optimization of the AMS-DDPG process by providing a distinctive and cutting-edge method for parameter optimization. By using ICWRS for parameter optimization, the model will be optimized and will function better in cloud-based systems.

Multi-Objective Optimization for Enhanced Work Scheduling

The suggested model has significant strength in the area of job scheduling, optimizing many multi-objective functions that are essential to cloud operations. These metrics include quality of service, makespan, energy consumption, CPU utilization, and migration costs. The complete strategy of the proposed model, which maximizes these complex features, represents a major advancement toward obtaining higher performance and more effective job scheduling in cloud computing settings.

Elevated Performance and Task Scheduling Rates: Experimental Validation

The suggested architecture has shown its ability to attain high performance and task scheduling rates in the cloud via an extensive experimental validation procedure. The comprehensive experimental validations were out on a variety of circumstances and workloads validate the effectiveness of the suggested model. The promising results validate this research's ability to greatly influence and progress cloud-based operations, opening the door for more study and use in actual cloud systems.

To sum up, our study has made significant contributions to the field by creating and validating a novel and comprehensive model that includes the ICWRS algorithm model, the AMS-DDPG method, and multi-objective optimization approaches. When combined, these elements provide a thorough answer to the problems associated with virtual machine migration and job scheduling, demonstrating the potential to improve cloud computing's effectiveness and performance. The promising experimental results highlight the importance of this work and provide new directions for further study and real-world implementation in the rapidly developing field of cloud computing.

References

[1] S. S. Gill et al., ‘Transformative effects of IoT, Blockchain and Artificial Intelligence on cloud computing: Evolution, vision, trends and open challenges’, Internet of Things, vol. 8, p. 100118, 2019.

[2] C. Tang, C. Zhu, X. Wei, H. Wu, Q. Li, and J. J. Rodrigues, ‘Intelligent resource allocation for utility optimization in rsu-empowered vehicular network’, IEEE Access, vol. 8, pp. 94453–94462, 2020.

[3] L. Helali and M. N. Omri, ‘A survey of data center consolidation in cloud computing systems’, Computer Science Review, vol. 39, p. 100366, 2021.

[4] B. Pourghebleh, A. Aghaei Anvigh, A. R. Ramtin, and B. Mohammadi, ‘The importance of nature-inspired meta-heuristic algorithms for solving virtual machine consolidation problem in cloud environments’, Cluster Computing, vol. 24, no. 3, pp. 2673–2696, 2021.

[5] Y. Huang, H. Xu, H. Gao, X. Ma, and W. Hussain, ‘SSUR: an approach to optimizing virtual machine allocation strategy based on user requirements for cloud data center’, IEEE Transactions on Green Communications and Networking, vol. 5, no. 2, pp. 670–681, 2021.

[6] Q. Liu et al., ‘Ekt: Exercise-aware knowledge tracing for student performance prediction’, IEEE Transactions on Knowledge and Data Engineering, vol. 33, no. 1, pp. 100–115, 2019.

[7] Ko, G. C., Lee, D. H., & Choi, Y. (2019). Particle Swarm Optimization for Energy-efficient Virtual Machine Placement in Hybrid Cloud Systems. Journal of Grid Computing, 17(4), 717-732.

[8] Shao, Y., Liu, J., & Wang, L. (2017). Ant Colony System-based VM Allocation for Resource Optimization in Cloud Data Centers. Future Generation Computer Systems, 67, 257-266.

[9] Zhang, B., Yu, Z., & Liu, Y. (2020). Q-learning-based VM Allocation in Cloud Data Centers with QoS Constraints. IEEE Transactions on Parallel and

Distributed Systems, 31(5), 1178-1192.

[10] Reji, R. V., & Selvakumar, A. (2019). Deep Deterministic Policy Gradient-based VM Consolidation and Migration in Cloud Data Centers. International Journal of Computer Applications, 182(13), 22-26.

[11] Wang, Y., Chen, Y., & Cheng, H. (2021). A DDPG-based Approach for VM Allocation in Multi-cloud Environment. Journal of Cloud Computing: Advances, Systems and Applications, 10(1), 1-16.

[12] Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... & Petersen, S. (2015). Human-level control through deep reinforcement learning. Nature, 518(7540), 529-533.

[13] Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., ... & Wierstra, D. (2016). Continuous control with deep reinforcement learning. arXiv preprint arXiv:1509.02971.

[14] Sutton, R. S., & Barto, A. G. (2018). Reinforcement learning: An introduction. MIT press.

[15] Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. arXiv preprint arXiv:1707.06347.

[16] Q. He and H. He, ‘A novel method to enhance sustainable systems security in cloud computing based on the combination of encryption and data mining’, Sustainability, vol. 13, no. 1, p. 101, 2020.

[17] L. Booth, I. Rakita, and Others, Introduction to corporate finance. John Wiley & Sons, 2020.

[18] H. Tabrizchi and M. Kuchaki Rafsanjani, ‘A survey on security challenges in cloud computing: issues, threats, and solutions’, The journal of supercomputing, vol. 76, no. 12, pp. 9493–9532, 2020.

[19] H. Hamill, K. Hampshire, S. Mariwah, D. Amoako-Sakyi, A. Kyei, and M. Castelli, ‘Managing uncertainty in medicine quality in Ghana: the cognitive and affective basis of trust in a high-risk, low-regulation context’, Social Science & Medicine, vol. 234, p. 112369, 2019.

[20] F. Cheng, Y. Huang, B. Tanpure, P. Sawalani, L. Cheng, and C. Liu, ‘Cost-aware job scheduling for cloud instances using deep reinforcement learning’, Cluster Computing, pp. 1–13, 2022.

[21] S. Gupta et al., ‘Efficient prioritization and processor selection schemes for heft algorithm: A makespan optimizer for task scheduling in cloud environment’, Electronics, vol. 11, no. 16, p. 2557, 2022.

[22] H. Ning, Y. Li, F. Shi, and L. T. Yang, ‘Heterogeneous edge computing open platforms and tools for internet of things’, Future Generation Computer Systems, vol. 106, pp. 67–76, 2020.

[23] S. Ramanathan, K. Kondepu, M. Razo, M. Tacca, L. Valcarenghi, and A. Fumagalli, ‘Live migration of virtual machine and container based mobile core network components: A comprehensive study’, IEEE Access, vol. 9, pp. 105082–105100, 2021.

[24] L. Megouache, A. Zitouni, and M. Djoudi, ‘Ensuring user authentication and data integrity in multi-cloud environment’, Human-centric Computing and information sciences, vol. 10, pp. 1–20, 2020.

[25] S. Asgari et al., ‘Hybrid surrogate model for online temperature and pressure predictions in data centers’, Future Generation Computer Systems, vol. 114, pp. 531–547, 2021.

[26] M. Haris and S. Zubair, ‘Mantaray modified multi-objective Harris hawk optimization algorithm expedites optimal load balancing in cloud computing’, Journal of King Saud University-Computer and Information Sciences, vol. 34, no. 10, pp. 9696–9709, 2022.

[27] T. S. Ayyarao et al., ‘War strategy optimization algorithm: a new effective metaheuristic algorithm for global optimization’, IEEE Access, vol. 10, pp. 25073–25105, 2022.

[28] D. Rossier, V. La Franca, T. Salemi, S. Natale, and C. T. Gross, ‘A neural circuit for competing approach and defense underlying prey capture’, Proceedings of the National Academy of Sciences, vol. 118, no. 15, p. e2013411118, 2021.

[29] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, ‘Orb-slam3: An accurate open-source library for visual, visual--inertial, and multimap slam’, IEEE Transactions on Robotics, vol. 37, no. 6, pp. 1874–1890, 2021.

[30] G. Panesar and R. Chadha, ‘Cloud Computing Virtual Machine Migration Methodology’, 12 2022, pp. 193–200.

[31] S. Lapp, K. Jablokow, and C. McComb, ‘KABOOM: an agent-based model for simulating cognitive style in team problem solving’, Design Science, vol. 5, p. e13, 2019.

[32] J. Zhang, A. Koppel, A. S. Bedi, C. Szepesvari, and M. Wang, ‘Variational policy gradient method for reinforcement learning with general utilities’, Advances in Neural Information Processing Systems, vol. 33, pp. 4572–4583, 2020.

[33] S. Huang and S. Ontañón, ‘A closer look at invalid action masking in policy gradient algorithms’, arXiv preprint arXiv:2006. 14171, 2020.

[34] L. Zhou, S. Leng, Q. Liu, and Q. Wang, 'Intelligent UAV swarm cooperation for multiple targets tracking', IEEE Internet of Things Journal, vol. 9, no. 1, pp. 743–754, 2021.

[35] E. C. Strinati and S. Barbarossa, '6G networks: Beyond Shannon towards semantic and goal-oriented communications', Computer Networks, vol. 190, p. 107930, 2021.

[36] D. T. Bui, P. Tsangaratos, P.-T. T. Ngo, T. D. Pham, and B. T. Pham, 'Flash flood susceptibility modeling using an optimized fuzzy rule based feature selection technique and tree based ensemble methods', Science of the total environment, vol. 668, pp. 1038–1054, 2019.

[37] Kanna, D. R. K. ., Muda, I. ., & Ramachandran, D. S. . (2022). Handwritten Tamil Word Pre-Processing and Segmentation Based on NLP Using Deep Learning Techniques. Research Journal of Computer Systems and Engineering, 3(1), 35–42. Retrieved from https://technicaljournals.org/RJCSE/index.php/journal/article/view/39

[38] Ahmed Abdelaziz, Machine Learning Approaches for Predicting Stock Market Volatility , Machine Learning Applications Conference Proceedings, Vol 3 2023.

[39] Dhabliya, D., Dhabliya, R. Key characteristics and components of cloud computing (2019) International Journal of Control and Automation, 12 (6 Special Issue), pp. 12-18.

[40] Carmen Rodriguez, Predictive Analytics for Disease Outbreak Prediction and Prevention , Machine Learning Applications Conference Proceedings, Vol 3 2023.

[41] Jang Bahadur Saini, D. . (2022). Pre-Processing Based Wavelets Neural Network for Removing Artifacts in EEG Data. Research Journal of Computer Systems and Engineering, 3(1), 43–47. Retrieved from https://technicaljournals.org/RJCSE/index.php/journal/article/view/40

[42] Dhabliya, D., Sharma, R. Cloud computing based mobile devices for distributed computing (2019) International Journal of Control and Automation, 12 (6 Special Issue), pp. 1-4.