Scalable Computing: Practice and Experience, ISSN 1895-1767, http://www.scpe.org
© 2024 SCPE. Volume 25, Issues 5, pp. 3701–3714, DOI 10.12694/scpe.v25i5.3111

OPTIMIZING TASK SCHEDULING: EXPLORING ADVANCED MACHINE LEARNING IN DEW-POWERED CLOUD ENVIRONMENTS

A. GANESH[*], K SREE DIVYA[†], CHINTHAKUNTA SASIKALA[‡], E. POORNIMA[§], NIDAMANURU SRINIVASA RAO[¶], A.V.L.N SUJITH[‖] AND G.RAMESH[**]

Abstract. Research into Dew computing environments has recently emerged as a result of the increasing prevalence and processing power of mobile and IoT devices. In these settings, even low-powered devices can share some of their computational resources with their neighbors. This paper proposes a novel approach to workflow scheduling in dew enabled cloud computing environment, called Deep Q-learning (DQN) + Chronological Geese Migration Optimization (CGMO). DQN is a deep learning-based method for scheduling workflows, while CGMO is a hybrid optimization algorithm that combines the chronological idea and the Wild Geese Migration Optimization (GMO) algorithm. The proposed approach aims to optimize multiple objectives, including predicted energy, Quality of Service (QoS), and resource usage, by scheduling workflows in the cloud. The approach also takes into account the current state of the Virtual Machine (VM) and the job. The assessment measures employed for DCGM include maximum QoS, minimum energy usage, and maximum resource utilization. The results show that DCGM achieved the highest QoS (0.865), lowest energy usage (0.0322), and highest resource utilization (1.000) compared to other approaches.

Key words: Dew computing, deep learning, task scheduling

1. Introduction. In recent years, resource-constrained devices have been under severe computational strain due to the meteoric rise of computationally expensive jobs in mobile applications. Smartphones and Internet of Things devices, unfortunately, typically fall short of such requirements. Dew computing is a solution that offers moving computationally expensive tasks to more capable (nearby) machines [1, 2, 3, 4, 5]. A device is considered nearby if it is on the same local network as another device. The concept is that you may use your phone to delegate tasks to your computer, saving battery life on both devices. However, learning how to efficiently divide tasks across adjacent devices is a major hurdle for Dew computing to be useful in practice. The issue of scheduling tasks in Dew settings is investigated in this paper. The term "Dew environment" refers to a network of interconnected gadgets. Different devices may have different storage capacities, sensor arrays, processing speeds, wireless connectivity options, and battery life. Furthermore, consumers may engage with various gadgets at various moments. All these considerations are necessary for efficient task distribution in a Dew setting. Existing solutions for distributing work in Dew settings adhere to human-designed policies. By adhering to a predetermined set of rules, these policies attempt to distribute the workload evenly across the devices. The Simple Energy-Aware Scheduler (SEAS) [6], the Batch Processing Algorithm (BPA) [7], and the Round Robin (RR) [8] are all examples of such schedulers. Due to their inflexibility, these approaches consistently produce bad decisions and wasteful resources when applied to a Dew context.

In this study, we suggest utilizing RL to figure out how to allocate tasks in a Dew ecosystem [9]. Learning optimal behavior by interaction with an environment is the focus of RL, a branch of AI that investigates how

[*]Department of CSE, Sri Venkateswara College of Engineering, Tirupati, Andhra Pradesh, India, (achari.ganesh@gmail.com)
[†]Department of Computer Science & Technology, Madanapalle Institute of Technology and Science, Madanapalle, Andhra Pradesh, India. (divya.kpn@gmail.com)
[‡]Department of Computer Science and Engineering, Srinivasa Ramanujan Institute of Technology (Autonomous), Ananthapuramu, Andhra Pradesh,India, (sasikalareddy27@gmail.com)
[§]Department of CSE (AI& ML), Gokaraju Rangaraju Institute of Engineering and Technology, Hyderabad, India (poornimacse561@gmail.com)
[¶]Department of CSE, Narsimha Reddy Engineering College, Secunderabad, Telangana State, India (rao75nidamanuru@gmail.com)
[‖]Department of CSE, Narsimha Reddy Engineering College, Secunderabad,Telangana State, India (sujeeth.avln@gmail.com)
[**]Department of CSE, Gokaraju Rangaraju Institute of Engineering & Technology, Hyderabad, India (ramesh680@gmail.com)

to create such creatures. Every action the agent takes results in some form of reward signal that it then tries to maximize. This is done by the agent learning from its past mistakes and adjusting its present policy (a mapping from observations to actions). Research in fields as diverse as robotics [10], conversational agents [11], and drug discovery [12] have all shown success using RL agents to address complicated decision-making problems powered by deep learning. We propose letting an RL agent figure out how to divide up work in a Dew system through experience. The use of RL for task distribution in Edge and Cloud computing has been investigated in previous research [13, 14, 15, 16]. However, no one has yet implemented RL on a Dew system. The unique difficulties of Dew computing cannot be compared to those of Cloud or Edge computing. Some gadgets in Dew computing, for instance, may experience battery drain or user interaction. Further, it has not been investigated in prior publications if RL agents may acquire rules that generalize well to novel contexts. Our results demonstrate that, when tested in novel scenarios, deep RL agents outperform state-of-the-art heuristic approaches in their ability to learn to offload tasks.

Workflow, a relatively new technology, has been widely used to keep tabs on the best apps' performance. In the context of scientific disciplines, a workflow is the collection of individual actions that are interconnected by data [7, 8]. Workflow applications have expanded in recent years to include domains as diverse as e-commerce, biology, astronomy, and physics. The workflow's tasks, in general, need time- and resource-intensive operations to be carried out effectively [9]. Multi-objective scheduling-based techniques are broken down into two distinct types: QoS-inhibited algorithms and QoS-optimization algorithms [10]. Several approaches rely on QoS constraints to transform this problem into a single-objective optimization problem [2]. In order to influence QoS, extensive procedures are often designed and implemented in widely disseminated large-scale evaluation settings. How to coordinate the jobs has been a focus of the inquiry for a while now [9, 11]. Workflow scheduling solely considered the amount of investigations based on price and timeliness. When asked about how to reduce costs without sacrificing quality of service, they often recommend something called multi-objective scheduling, which is used to derive the pricing. The precaution was included into the workflow scheduling of certain recent studies based on cloud infrastructure. The impact of communications between tasks and virtual machines on cloud security as a whole was not adequately examined [3].

The allocation of jobs to relevant resources in CC [12] may either be insufficient or optimized to meet the needs of users depending on QoS. Cost, time, security, load, and success rate are all integrated into the QoS requirement based on process scheduling [13]. Nonetheless, many studies in this field have only optimized a few or a few QoS factors. A great deal of planning, plans, and improvements went into the framework of cloud security, all of which work together to guarantee the safety of data stored in cloud architectures [12]. The researchers developed many workflow scheduling strategies in which the technical word was associated with CC [5]. Due to the importance of the applications, a large number of studies have been conducted to develop a model for workflow management in clouds in line with scheduling schemes. These studies include Condor Dagman [14], Gridbus toolkit [15], Iceni [16], and Pegasus [17]. By hiding their orchestrations and implementations, the aforementioned structures may be seen as a kind of platform service that aids the network and cloud-based computerization of technical and commercial applications [4]. Workflow scheduling uses data mining and regression based methods. In addition, researchers often try new approaches to workflow scheduling in an attempt to fix the issues that have plagued the ones they've used before. Due to its easy convergence qualities, flexibility, and error-tolerance abilities, the Firefly algorithm stands out as the best of all the swarm-based models used in workflow scheduling [5].

In dew-powered cloud environments, efficiently scheduling microservice-enabled tasks is crucial for optimizing resource utilization and minimizing energy consumption. Traditional scheduling methods often fall short in adapting to the dynamic nature of these environments. To address this challenge, we propose a novel machine learning algorithm tailored for task scheduling. Our algorithm leverages advanced predictive models to dynamically allocate resources based on workload patterns and environmental conditions, thereby improving system performance and energy efficiency. This paper presents a comprehensive discussion on the applicability and benefits of our approach, bridging the gap between theoretical concepts and practical implementation in dew-powered cloud environments. In this paper, we provide a new framework, DQN+CGMO, based on multi-objective and DL for scheduling cloud-based workflows. In this study, we focus on two processes: multi-objective workflow scheduling and workflow scheduling using DL. Predicted energy, QoS, and resource consumption, actual task running time, bandwidth utilization, memory capacity, makespan equivalent of total cost, and task priority are all taken into account to determine the fitness of workflow scheduling in multi-objective-based workflow scheduling. Workflow scheduling is done with consideration for CGMO, which is derived from the combination of the GMO algorithm with the chronological idea. At the same time, DQN is fed the incoming task in the DL-based workflow scheduling process. Workflow scheduling also takes into account VM and task settings in real time. In the end, the combined product is what we get.

2. Literature Review. The concept of EMO (Evolutionary Multi-objective Optimization) was created by Zhu, Z., et al. [2]. This approach was motivated by the Amazon EC2 on-demand instance kinds. If only we could simply include IaaS's resourcing choices and cost estimate tools into the suggested paradigm! The term "regressive" was coined by the researchers G. Narendrababu Reddy and S. Phani Kumar [5]. RWO optimization (Whales optimization). The RWO model maximized resource efficiency with minimum expenditure of time, money, and energy. The original presentation of the Whale optimization algorithm (WOA) may be found in the work of Strumberger (I.), et al. [18]. In this work, we improve the implementation of the unique whale optimization and explore various approaches to solving CC's resource scheduling problem. Stephanakis, I.M., et al. [19] developed particle swarm optimization (PSO). It was a dividing line in the scheme's design, and it helped many individuals in the population agree on the best course of action.

A new paradigm called as Mobile Edge Computing has arisen to deal with the latency and network traffic problems that plague Mobile Cloud Computing systems [23]. Khan et al. [24], stating that "Edge Computing" refers to "a model that allows a cloud-based computing capacity providing services making use of the infrastructure that is on the edge of the network." Mobile Edge Computing is the practice of deploying more robust applications by using local servers or workstations inside a network to minimize latency and maximize data processing efficiency. This paradigm allows for the implementation of more reliable and efficient applications [25, 26] by enabling Edge servers to collaborate with neighboring nodes or with Cloud services. Despite the fact that Edge computing helps to ease network challenges, the network backbone may not be accessible or available while working with IoT devices in, say, mines and on ships, in deserts, or when traveling.

In dew computing, nearby machines on a network take turns processing data. This suggested design [27] aims to reduce network latency, the energy cost of transferring data over great distances, and the expenses associated with utilising Cloud infrastructure. Dew computing does this by enhancing functionality in two fundamental areas of mobile and IoT device performance. To begin, it treats mobile devices as clients inside the network architecture, offloading their duties to other devices on the same network. Second, in Dew computing, mobile and IoT devices are seen as resources that may be exploited to improve the performance of the existing system. This method enables one device in the network to use the capabilities of another (including mobile and IoT devices) to accomplish a goal [29,30, 31]. Ramesh et al. [32] explored machine learning algorithms with conventional network defense mechanisms offers a proactive and flexible strategy to protect systems from Distributed Denial of Service (DDoS) attacks, ensuring robust security measures that can adapt to evolving threats in real-time.

Jia, Y.H., et al. [9] developed the method now known as Ant Colony Optimization (ACO). This strategy received a flawless score in both the cost and efficiency categories. Time was lost because precise methods were not documented, even though they were used by numerous workflow programs. F. Abazari et al. created a method called multi-objective workflow scheduling (MOWS). The strategy's better security and risk in a wide range of workload characteristics led to an overall improvement in the building's integrity. Additionally, it demonstrated that the developed system successfully predicted attacks in cloud environments. A significant advancement in both cost and energy utilization was achieved by the development of Dynamic Voltage and Frequency Scaling with Multi-objective Discrete Particle Swarm Optimization (DVFS-MODPSO) by Yassa, S., et al. [4]. However, issues of trust and safety were not addressed. Kakkottakath Valappil Thekkepuryil et al. presented Ant Lion optimization (ALO) in [12]. ALO had a great convergence rate and was very searchable. However, there was a lack of multi-cloud collaboration integration in various clouds, therefore just one service was provided.

3. System Model. With the help of CC, the on-demand services are furnished on the basis of the necessities of user where the operation will be conducted. Therefore, numerous applications of workflow in CC may be operated with the available on-demand is the critical task in the scheduling of workflow. An

Table 2.1: Literature Analysis

Optimization Technique	Origin	Motivation/Application	Limitations/Gaps
Evolutionary Multi-objective Optimization (EMO)	Zhu, Z., et al. [2]	Inspired by Amazon EC2 on-demand instance kinds; aims to optimize resourcing choices and cost estimates in cloud environments.	Limited documentation on precise methods used; potential gap in methodological transparency.
Whale Optimization Algorithm (WOA)	Strumberger (I.), et al. [18]	Focuses on maximizing resource efficiency with minimal expenditure of time, money, and energy; applied to solve cloud computing's resource scheduling problem.	Lack of consideration for multi-cloud collaboration integration; limited to single-service provisioning.
Particle Swarm Optimization (PSO)	Stephanakis, I.M., et al. [20]	Facilitates collaboration among individuals in the population to determine the best course of action; used for optimization tasks in cloud environments.	May face challenges in handling complex optimization landscapes; potential for premature convergence.
Ant Colony Optimization (ACO)	Jia, Y.H., et al. [9]	Achieves high scores in cost and efficiency categories; applied to workflow scheduling problems in cloud environments.	Limited documentation on the precise implementation details; potential for replication issues.
Multi-objective Workflow Scheduling (MOWS)	F. Abazari et al.	Improves security and risk management while enhancing building integrity; effectively predicts attacks in cloud environments.	Limited consideration for trust and safety aspects; potential gaps in addressing security vulnerabilities.
Dynamic Voltage and Frequency Scaling with Multi-objective Discrete Particle Swarm Optimization (DVFS-MODPSO)	Yassa, S., et al. [4]	Enhances cost and energy utilization efficiency; addresses challenges in cloud computing resource management.	Does not address trust and safety concerns; potential limitations in real-world scalability and applicability.
Ant Lion Optimization (ALO)	Thekkepuryil et al.	Demonstrates high convergence rate and searchability; primarily focused on optimization tasks with single-cloud services.	Limited integration for multi-cloud collaboration; potential gaps in addressing diverse service requirements.

effectual scheduler should be required for the proper scheduling with the consideration of available resources. In addition to that, in order to process the sensitive data to control the safety and reliability of data sharing is the significant part in the secure infrastructure. Hence, the finest scheduling of workflow in the CC is designed with the advancement of safety measures that is illustrated in figure 3.1. The pre-processing data in CC has physical machines (PM) and VM, which offers the service to the user on the basis of demand. the scheduling of workflow is utilized finely in the presented model on the basis of multi-objective function in accordance with six attributes namely, resource utilization, QoS, bandwidth utilization, makespan equivalent of total cost, predicted energy, and memory capacity.

Consider the PMs employed in CGMO is illustrated by,

$$P = \{P_1, P_2, ...P_u, ..., P_v\} \tag{3.1}$$

The Directed Acyclic Graph (DAG) is employed for the illustration of the scheduling of workflow and it is expressed as Q = (G, L) . The factor G signifies the task equivalent to the workflow that is computed as,

$$G = \{G_1, G_2, ...G_K, ..., G_J\} \tag{3.2}$$

4. Proposed DQN+CGMO. The fundamental purpose of this study is to develop DQN+CGMO, a multi-objective and DL-based hybrid optimization for workflow scheduling in CC. In this setup, CGMO sched-

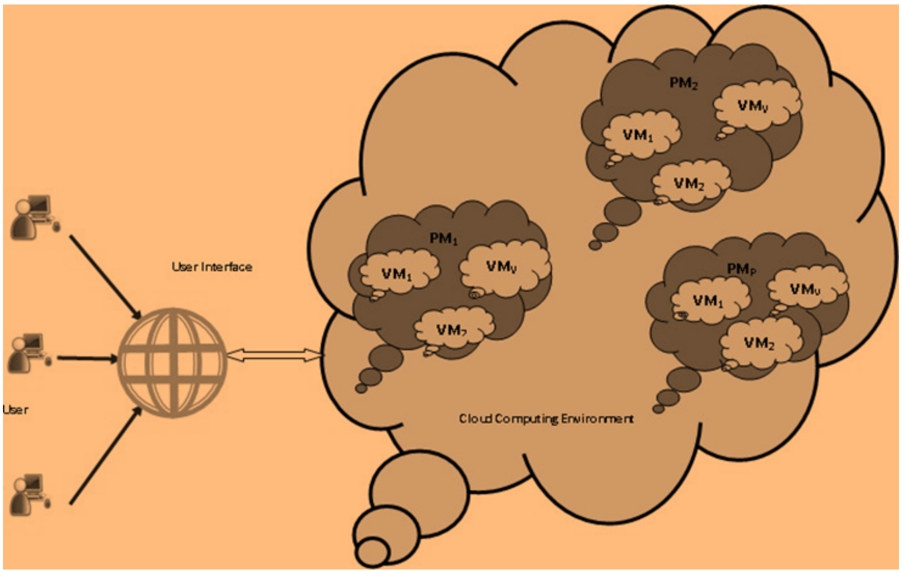

Fig. 3.1: Systematic View of workflow scheduling in cloud within the interval.

ules workflows based on many objectives, whereas DQN schedules workflows using deep learning. Predicted energy, QoS, and resource consumption, actual task running time, bandwidth utilization, memory capacity [20], makespan equivalent of total cost [21], and task priority are all used to determine the fitness of workflow scheduling in multi-objective-based workflow scheduling. CGMO, a hybrid of the GMO algorithm and the chronological idea, is used to schedule workflows [22]. The incoming job is also supplied to DQN [23] in the DL-based workflow scheduling process. Workflow scheduling also takes into account real-time VM and task characteristics. Here, both the tasks' parameters, such as average computation cost, earliest start time, earliest finish time, duration, and priority, and the VMs' parameters, including bandwidth utilization, memory utilization, capacity, and central processing unit (CPU), are taken into account. Last but not least, the total production is what matters most. Schematic representation of CGMO for DQN-based multi-objective workflow scheduling is shown in Figure 2. Combining Deep Q-learning (DQN) with Chronological Geese Migration Optimization (CGMO) could yield an innovative approach for optimizing task scheduling in dew-powered cloud environments. Here are the steps involved in this hybrid algorithm:

Initialization

- Set up the environment with historical data on task execution, resource utilization, and environmental conditions.
- Initialize the DQN and CGMO components of the algorithm.

Data Preprocessing

- Normalize and preprocess the input data to ensure consistency and remove noise.
- Feature engineering may be applied to extract relevant information, such as workload patterns and environmental factors.

Deep Q-learning (DQN) Exploration

- Utilize the DQN component to explore the state-action space and learn optimal task scheduling policies.
- Train the DQN model using historical data to estimate the Q-values, which represent the expected future rewards for taking specific actions in given states.
- Employ techniques such as experience replay and target network updates to stabilize training and improve convergence.

Chronological Geese Migration Optimization (CGMO)

- Implement the CGMO component inspired by the behavior of geese migrating in chronological order.

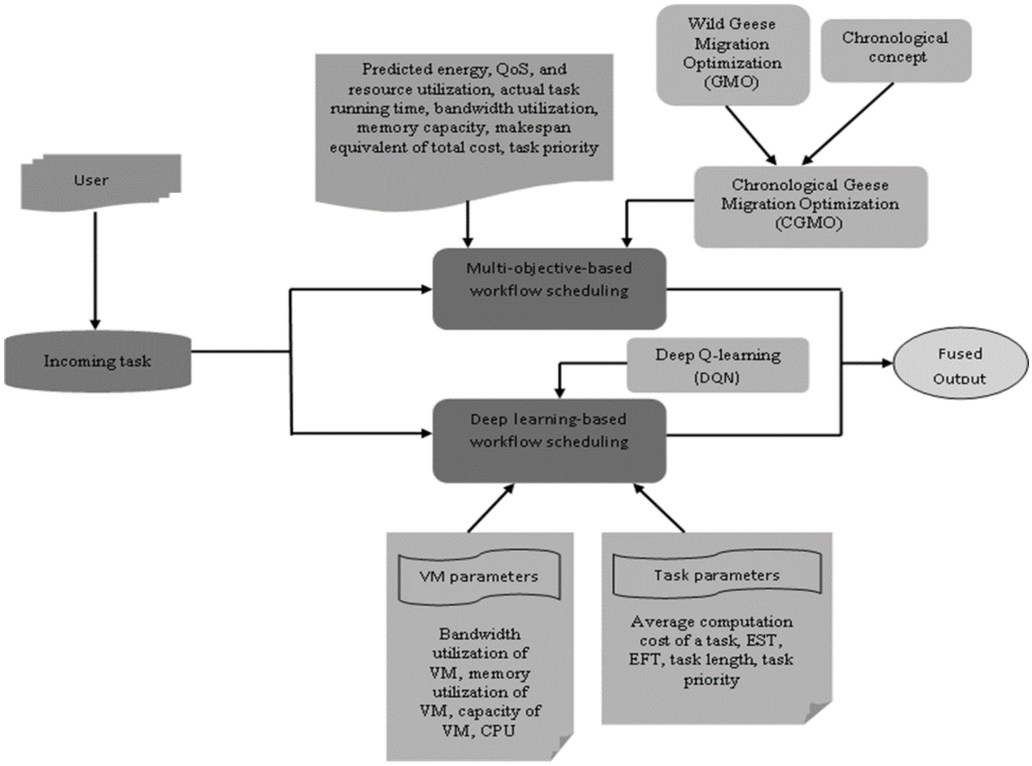

Fig. 4.1: Modeled view of CGMO for multi-objective and DQN based workflow scheduling

- Define migration paths as potential solutions to the task scheduling problem, where each path represents a sequence of task assignments over time.
- Use CGMO to iteratively update and optimize the migration paths based on historical patterns and environmental conditions.
- Leverage the collective intelligence of the population of "geese" (i.e., candidate solutions) to guide the search towards promising regions of the solution space.

Hybridization and Policy Integration

- Integrate the learned policies from the DQN with the optimized migration paths generated by CGMO.
- Combine the strengths of both algorithms to adaptively adjust task scheduling decisions in response to changing conditions and uncertainties.

Evaluation and Fine-Tuning

- valuate the hybrid algorithm's performance using simulation or real-world experiments.
- Fine-tune the algorithm parameters and hyperparameters based on performance metrics such as resource utilization, energy efficiency, and task completion time.

Deployment and Continuous Learning

- Deploy the optimized scheduling algorithm in the dew-powered cloud environment.
- Monitor system performance and collect feedback data to facilitate continuous learning and adaptation.
- Update the algorithm periodically to accommodate evolving workload patterns and environmental dynamics.

4.1. Multi-objective based task scheduling. The system of task scheduling is significant for the enhancement of resources and the effectualness of server and also by improving the assessment of the analyzed nodes. In the multi-objective task set, the numerous tasks of multi-objective are defined time during the time processing of CC. In accordance with the execution time, the entire tasks of multi- objective on VM evaluate

Pseudo Code of CGMO

```
Input: Population size M , maximum iteration Max_iter
Output: A_j^{r+1}
Begin
Initialize the population
for r = 1: Max_iter
for n = 1: K
for j = 1: M
Generate the migration groups using Eq. [17]
end for
end for
if rand > 0.5
for n = 1: K
for ( j = a *(n -1)+1: (a * n))
Upgrade the new solution employing Eq. [24] and Eq. [25]
end for
end for
else
for n =: K
for ( j = a *(n -1)+1: (a * n))
Evaluate free foraging utilizing Eq. [26]
end for
end for
else
Compute the radius of migration group by Eq. [27]
end for
Return
Terminate
```

the load balance difference of task scheduling of multi-objective and create the intent function of CC scheduling of multi-objective task.

4.2. Deep Learning based task scheduling. Recurrent Neural Networks (RNNs) and long short-term memories (LSTMs) (RNN-LSTMs) are used to process incoming tasks for DL-based task scheduling. VM and task settings will determine which job is chosen. Let's assume for the time being that the task's computing cost, EST, EFT, duration, and priority are all fixed. Memory, CPU, and ram are the virtual machine settings.

5. Comparative Discussion. In this section, the effectiveness of the provided technique is shown in relation to earlier models. Maximum Qos (0.785), minimum energy usage (0.022), and maximum resource utilization (1.000) were all reached using the assessment methods used for DQN+CGMO. Table 5.1 presents the results of a comparative analysis of DQN and CGMO.

5.0.1. Deep Learning based task scheduling. Recurrent Neural Networks (RNNs) and long short-term memories (LSTMs) (RNN-LSTMs) are used to process incoming tasks for DL-based task scheduling. VM and task settings will determine which job is chosen. Let's assume for the time being that the task's computing cost, EST, EFT, duration, and priority are all fixed. Memory, CPU, and ram are the virtual machine settings.

In DQN [23] the basic procedure is to allow the initial state to the Neural network (NN). Moreover, NN precedes the entire tasks at the output stage. When the current value and deep-TD target values are similar, the upgrade regulations of

J(Z,ψ) is not able to upgrade the values. Therefore, J(Z,ψ) congregates to the original values and observes the preferred objective.

6. Results and Discussion. This section discusses not just the end result of DQN+CGMO, but also the experimental setup, simulation settings, metrics used, and comparative evaluation. The quality of service (QoS), energy consumption (EC), and resource utilization (RU) are the metrics used by the proposed DCGM

Table 5.1: Comparative Discussion (*DQN+CGMO)

Dataset	Metrics	RWWO_DMN+DRL	RWO +DRL	WOA +DRL	PSO+ DRL	ACO +DRL	RWWTDO+DRL	Proposed*
PM=5, VM=10	QoS	0.174	0.273	0.501	0.529	0.574	0.673	0.750
PM=5, VM=10	Energy Consumption	0.485	0.399	0.043	0.040	0.039	0.038	0.035
PM=5, VM=10	Resource utilization	0.947	0.960	0.961	0.964	0.974	0.975	0.979
PM=10, VM=15	QoS	0.179	0.279	0.526	0.538	0.579	0.679	0.761
PM=10, VM=15	Energy Consumption	0.489	0.405	0.045	0.040	0.037	0.0297	0.022
PM=10, VM=15	Resource utilization	0.953	0.961	0.963	0.975	0.977	0.980	0.989
PM=15, VM=20	QoS	0.194	0.294	0.525	0.550	0.589	0.693	0.771
PM=15, VM=20	Energy Consumption	0.507	0.408	0.047	0.041	0.038	0.030	0.021
PM=15, VM=20	Resource utilization	0.963	0.968	0.970	0.971	0.980	0.983	0.987
PM=20, VM=25	QoS	0.204	0.305	0.529	0.565	0.599	0.704	0.785
PM=20, VM=25	Energy Consumption	0.508	0.420	0.048	0.042	0.040	0.031	0.22
PM=20, VM=25	Resource utilization	0.968	0.997	0.997	0.998	1.000	1.000	1.000

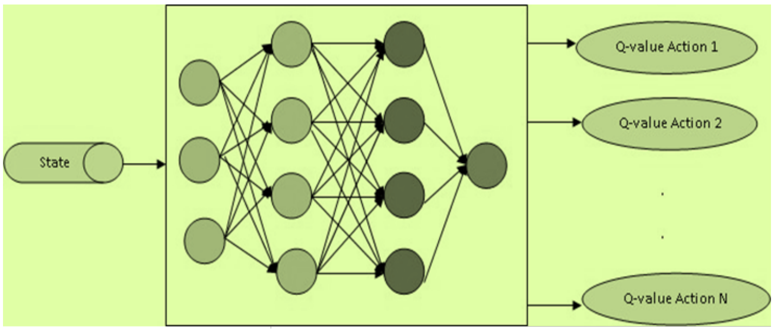

Fig. 5.1: Structure of DQN

method. The RWWO_ DMN+DRL [27] [28], RWO+DRL[5], WOA+DRL[18], PSO+DRL[19], ACO+DRL[9], and RWWTDO+DRL [3] comparison methods are used for DQN+CGMO.

The assessment of DCGM for varied job sizes with PM=5, VM=10, is shown in Figure 6.1,6.2,6.3. The DCGM based on QoS are discussed in Figure 6.1. While the previous approaches, namely RWWO_ DMN+DRL, RWO+DRL, WOA+DRL, PSO+DRL, ACO+DRL, and RWWTDO+DRL, obtained 0.174, 0.273, 0.501, 0.529, 0.574, and 0.673, the DCGM achieved the QoS of 0.750 when the task size=400. Figure 6.2 shows how much energy DCGM uses. The classic techniques such as RWWO_ DMN+DRL had 0.485, RWO+DRL had 0.399, WOA+DRL had 0.043, PSO+DRL had 0.040, AC+DRL had 0.039, and RWWTDO+DRL had 0.038 with 400 as the energy consumption if the DCGM obtained 0.035. Figure 6.3 explains the DCGM in terms of resource consumption. The traditional techniques achieved 0.947, 0.960, 0.961, 0.9640.974, 0.975 for RWWO_ DMN+DRL, RWO+DRL, WOA+DRL, PSO+DRL, ACO + DRL, and RWWTDO +DRL, while the DCGM earned 0.979 based on resource utilization with a work size of 400.

6.1. Assessment of DCGM based on task size with PM=20, VM=25. Figure 6.4,6.5,6.6 shows the worth of DCGM for different work sizes (PM=20, VM=25). Figure 6.4 depicts the assembly of a DCGM supporting QoS. When the task size was 400, the gains for the conventional methods were 0.204, 0.305, 0.529, 0.565, 0.599, and 0.704, respectively. The DCGM reached an all-time high of 0.785. The use of energy by DCGM is shown in Figure 6.5. For comparison, the energy consumption results for the RWWO_ DMN+DRL, RWO+DRL, WOA+DRL, PSO+DRL, ACO+DRL, and RWWTDO+DRL were 0.508, 0.420, 0.048, 0.042, 0.040, and 0.031, respectively, for a task size of 400. The DCGM achieved 0.022. Figure 6.6 depicts the typical approaches next to the DCGM with resource utilization at 1.000, including RWWO_ DMN+DRL = 0.968, RWO+DRL = 0.997, WOA+DRL = 0.997, PSO+DRL = 0.998, ACO+DRL = 1.000, and RWWTDO+DRL

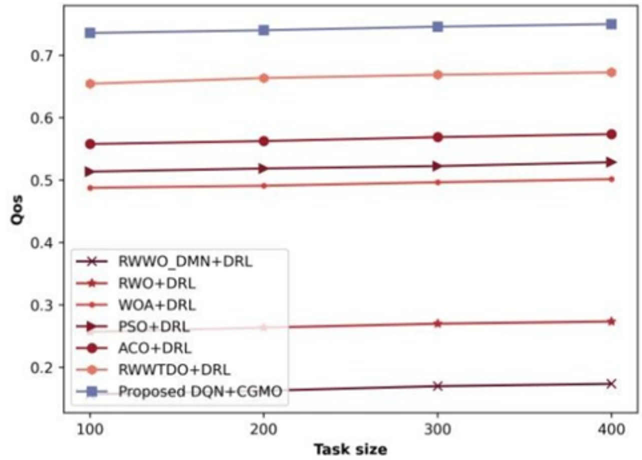

Fig. 6.1: Valuation of DCGM altering task size with PM=5, VM=10 QoS

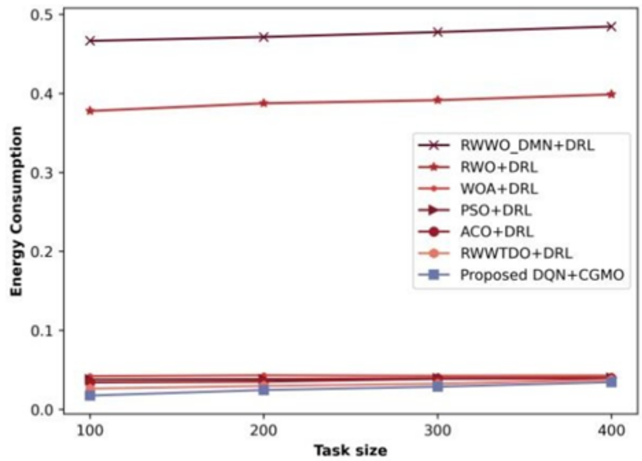

Fig. 6.2: Valuation of DCGM altering task size with PM=5, VM=10 Energy Consumption

= 1.000.

7. Comparative Discussion. In this section, the effectiveness of the provided technique is shown in relation to earlier models. Maximum Qos (0.785), minimum energy usage (0.022), and maximum resource utilization (1.000) were all reached using the assessment methods used for DQN + CGMO. Table 1 presents the results of a comparative analysis of DQN and CGMO.

7.1. Applications of Proposed algorithm in Real-world environment: An illustrative Scenario. Envision a sophisticated city infrastructure with numerous IoT sensors used to monitor several elements of urban life, including traffic flow, air quality, and energy use. Various technologies, including as sensors on lampposts and smart meters in residences, produce large volumes of data that must be processed and evaluated immediately to allow city officials to make well-informed decisions. Dew computing is a promising approach that utilizes the computational capabilities of IoT devices to analyze data closer to the source, hence decreasing latency and bandwidth usage. Scheduling workflows effectively in a changing context is a major difficulty. Introduce the suggested method, DQN + CGMO, which aims to optimize workflow scheduling in Dew-enabled cloud computing settings. Let's examine a real-life situation: To enhance traffic flow by evaluating live data

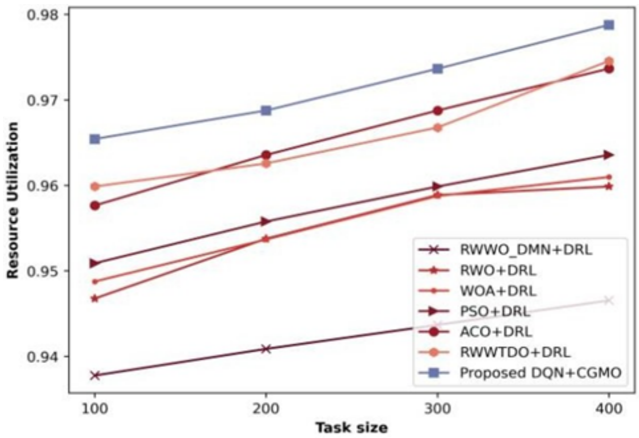

Fig. 6.3: Valuation of DCGM altering task size with PM=5, VM=10 Resource utilization

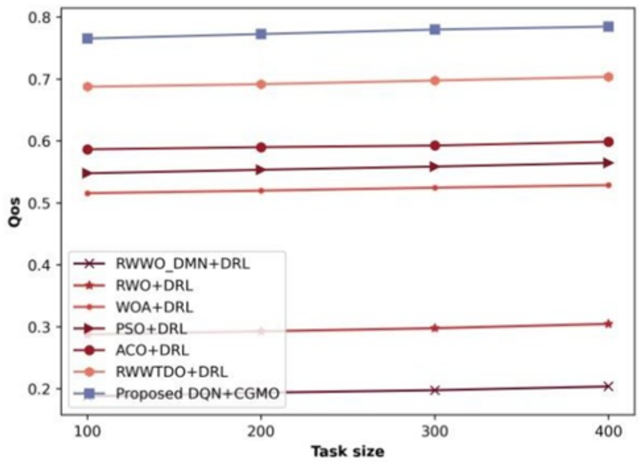

Fig. 6.4: Valuation of DCGM altering task size with PM=20, VM=25 QoS,

from traffic cameras, vehicle sensors, and GPS systems in automobiles.

- Data Collection: Traffic data is gathered from a variety of IoT devices located across the city, such as traffic cameras, car sensors, and GPS devices.
- Data Processing: The gathered data must undergo processing to identify traffic congestion, recognize traffic trends, and forecast traffic flow in various metropolitan locations.
- Workflow scheduling utilizes the DQN + CGMO algorithm. It efficiently organizes processes by taking into account aspects like anticipated energy use, Quality of Service (QoS), and resource usage.
- Implementation: Workflows are carried out using a distributed network consisting of IoT devices, cloud servers, and edge computing nodes. Tasks are assigned in real-time according on the devices' present status and the workload.
- Feedback and Optimization: The system continuously adjusts to evolving traffic circumstances and device capabilities. It modifies the schedule of workflow to provide the best performance and efficiency.

The proposed hybrid algorithm combining Deep Q-learning (DQN) with Chronological Geese Migration Optimization (CGMO) holds significant potential for addressing various challenges in real-world industry settings. Let's explore some concrete examples of its applications across different domains:

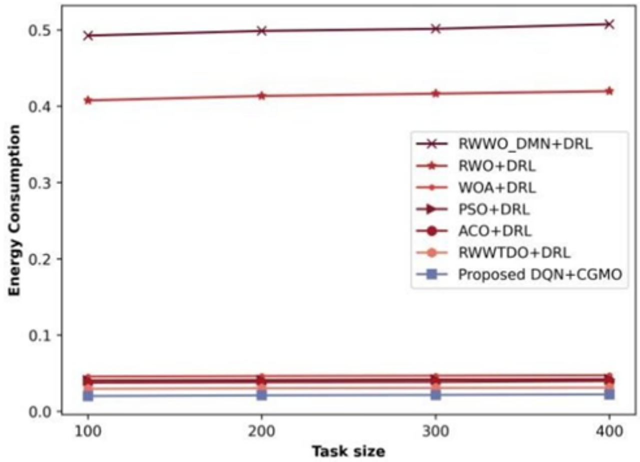

Fig. 6.5: Valuation of DCGM altering task size with PM=20, VM=25 b) Energy Consumption

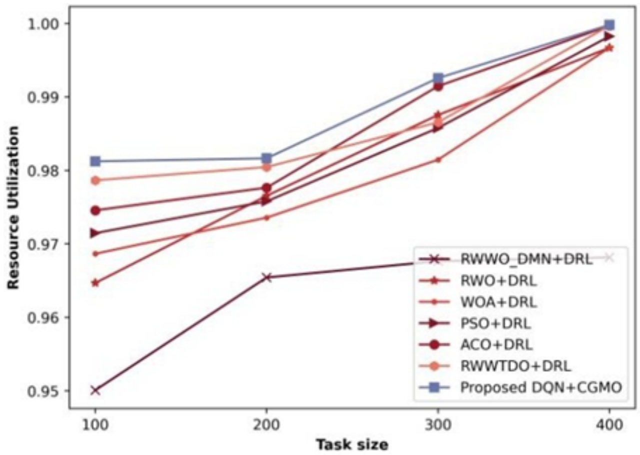

Fig. 6.6: Valuation of DCGM altering task size with PM=20, VM=25 Resource utilization

7.1.1. Cloud Computing and Data Centers.

- Dynamic Resource Allocation: In cloud computing environments, where resource demands fluctuate frequently, the hybrid algorithm can dynamically allocate resources based on workload patterns and environmental conditions. For example, in a data center serving multiple clients with varying processing requirements, the algorithm can optimize resource utilization by intelligently scheduling tasks on available servers, ensuring efficient use of computing resources while minimizing energy consumption.
- Fault Tolerance and Load Balancing: The algorithm can also enhance fault tolerance and load balancing in distributed systems. By continuously monitoring the health and performance of servers, it can redistribute tasks in response to failures or overloads, ensuring high availability and reliability of services. For instance, in a web hosting environment experiencing sudden spikes in traffic, the algorithm can automatically scale resources and distribute incoming requests across servers to maintain optimal performance.

7.1.2. Manufacturing and Supply Chain Management.

- Production Scheduling: In manufacturing facilities, the algorithm can optimize production schedules by intelligently allocating resources (e.g., machines, workers) to different tasks based on demand forecasts and production constraints. For example, in a manufacturing plant producing automotive parts, the algorithm can dynamically adjust production schedules in response to changes in demand, inventory levels, and machine availability, optimizing throughput and minimizing production costs.

- Inventory Management: The algorithm can also improve inventory management in supply chain networks by optimizing order fulfillment and logistics operations. For instance, in a retail distribution center managing inventory across multiple warehouses, the algorithm can optimize the routing and scheduling of delivery trucks to minimize transportation costs and ensure timely delivery of goods to customers.

7.2. Implications and Potential Limitations of Proposed Work.

7.2.1. Implications.

- Improved Efficiency and Resource Utilization: The hybrid algorithm has the potential to significantly enhance efficiency and resource utilization in cloud environments by dynamically optimizing task scheduling decisions based on real-time data and environmental conditions. This can lead to cost savings, energy efficiency improvements, and better overall performance.
- Enhanced Scalability and Flexibility: By leveraging the capabilities of both DQN and CGMO, the algorithm can adapt to a wide range of workload patterns and system dynamics, making it well-suited for highly scalable and flexible cloud environments. It can handle diverse workloads and dynamically adjust scheduling policies to accommodate changing demands and resource availability.
- Better Resilience and Fault Tolerance: The algorithm's ability to dynamically adapt to changing conditions and redistribute tasks in response to failures or overloads can enhance system resilience and fault tolerance. It can help mitigate the impact of hardware failures, network outages, or sudden spikes in demand by quickly reallocating resources and rerouting tasks to unaffected components.

7.2.2. Potential Limitations.

- Complexity and Computational Overhead: Implementing and fine-tuning the hybrid algorithm may require significant computational resources and expertise, particularly in training the DQN model and optimizing CGMO parameters. The algorithm's complexity could pose challenges in real-time decision-making and scalability, especially in large-scale cloud environments with high workload variability.
- Sensitivity to Environmental Factors: The algorithm's performance may be influenced by the accuracy and reliability of environmental data, such as temperature, humidity, and renewable energy availability. Inaccurate or noisy data could lead to suboptimal scheduling decisions, highlighting the importance of robust data preprocessing and quality assurance mechanisms.

7.2.3. Evolution and Adaptability.

- Continuous Learning and Optimization: The hybrid algorithm can evolve over time through continuous learning and optimization, leveraging feedback data to refine its decision-making policies and adapt to evolving workload patterns and system dynamics. By incorporating reinforcement learning techniques, the algorithm can autonomously improve its performance and adaptability over time without manual intervention.
- Customization for Different Cloud Environments: The algorithm can be customized and fine-tuned to meet the specific requirements and characteristics of different cloud environments, such as public clouds, private clouds, and hybrid clouds. By adjusting model parameters, optimization objectives, and decision-making criteria, the algorithm can adapt to diverse deployment scenarios and optimize performance in various operational contexts.

8. Conclusion. DCGM is an innovative approach for scheduling workflow in CC that uses multi-objective and DL. This study incorporates two processesmulti-objective workflow scheduling using DL, and vice versa. The fitness of workflow scheduling is measured using nine criteria in multi-objective scheduling. CGMO, which is derived from the combination of the GMO algorithm with the chronological idea, is taken into account throughout the workflow scheduling process. At the same time, DQN is fed the incoming task in the DL-based workflow scheduling process. Workflow scheduling also takes into account VM and task settings in real time.

Here, both the task parameters (such as the average computing cost of a task, the EST of a task, the EFT of a job, the duration of a task, and its priority) and the VM characteristics (such as the VM's bandwidth usage, memory utilization, capacity, and central processing unit) are taken into account. At the completion of this operation, the two outputs are combined. Maximum Qos (0.785), minimum energy usage (0.022), and maximum resource utilization (1.000) were all reached using the assessment methods used for DQN+CGMO. The study will be expanded to include additional hybrid networks, and the suggested method will eventually be included with cloud toolkits.

REFERENCES

[1] Mell, P., and Grance, T., *The NIST definition of cloud computing*,2011.

[2] Zhu, Z., Zhang, G., Li, M. and Liu, X., *Evolutionary multi-objective workflow scheduling in cloud*, IEEE Transactions on parallel and distributed Systems, vol. 27, no. 5, pp.1344-1357, 2015.

[3] Abazari, F., Analoui, M., Takabi, H. and Fu, S., *MOWS: multi-objective workflow scheduling in cloud computing based on heuristic algorithm*, Simulation Modelling Practice and Theory, vol. 93, pp.119-132, 2019.

[4] Yassa, S., Chelouah, R., Kadima, H. and Granado, B., *Multi-objective approach for energy-aware workflow scheduling in cloud computing environments*",The Scientific World Journal, 2013.

[5] Narendrababu Reddy, G. and Phani Kumar, S., *Regressive whale optimization for workflow scheduling in cloud computing*, International Journal of Computational Intelligence and Applications", vol. 18, no. 04, pp.1950024, 2019.

[6] Masdari, M., ValiKardan, S., Shahi, Z. and Azar, S.I., *Towards workflow scheduling in cloud computing: a comprehensive analysis*, Journal of Network and Computer Applications, vol. 66, pp.64-82, 2016.

[7] Chen, W.N. and Zhang, J., *An ant colony optimization approach to a grid workflow scheduling problem with various QoS requirements*, IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 39, no. 1, pp.29-43, 2008.

[8] Abrishami, S., Naghibzadeh, M. and Epema, D.H.,*Deadline-constrained workflow scheduling algorithms for infrastructure as a service clouds* Future generation computer systems, vol. 29, no. 1, pp.158-169, 2013.

[9] Jia, Y.H., Chen, W.N., Yuan, H., Gu, T., Zhang, H., Gao, Y. and Zhang, J., *An intelligent cloud workflow scheduling system with time estimation and adaptive ant colony optimization*, IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 1, pp.634-649, 2018.

[10] Arabnejad, H. and Barbosa, J.G., *A budget constrained scheduling algorithm for workflow applications*, Journal of grid computing, vol. 12, pp.665-679, 2014.

[11] Ludäscher, B., Altintas, I., Berkley, C., Higgins, D., Jaeger, E., Jones, M., Lee, E.A., Tao, J. and Zhao, Y., *Scientific workflow management and the Kepler system*, Concurrency and computation: Practice and experience, vol. 18, no. 10, pp.1039- 1065, 2006.

[12] Kakkottakath Valappil Thekkepuryil, J., Suseelan, D.P. and Keerikkattil, P.M., *An effective meta-heuristic based multi-objective hybrid optimization method for workflow scheduling in cloud computing environment*, Cluster Computing, vol. 24, pp.2367-2384, 2021.

[13] Kaur, P. and Mehta, S., *Resource provisioning and work flow scheduling in clouds using augmented Shuffled Frog Leaping Algorithm*, Journal of Parallel and Distributed Computing, vol. 101, pp.41-50, 2017.

[14] Thain, D., Tannenbaum, T. and Livny, M., *Condor and the Grid. Grid computing: Making the global infrastructure a reality*, pp.299-335, 2003.

[15] Buyya, R. and Venugopal, S., *The gridbus toolkit for service oriented grid and utility computing: An overview and status report*, In 1st IEEE International Workshop on Grid Economics and Business Models, GECON, pp. 19-66, IEEE, April 2004.

[16] McGough, S., Young, L., Afzal, A., Newhouse, S. and Darlington, J., *Workflow enactment in ICENI*, In UK e-Science All Hands Meeting, vol. 9, pp. 894-900, September 2004.

[17] Deelman, E., Blythe, J., Gil, Y., Kesselman, C., Mehta, G., Patil, S., Su, M.H., Vahi, K. and Livny, M., *Pegasus: Mapping scientific workflows onto the grid* In Grid Computing: Second European AcrossGrids Conference, AxGrids 2004, Nicosia, Cyprus, January 28-30, 2004. Revised Papers, pp. 11-20, Springer Berlin Heidelberg, 2004.

[18] Strumberger, I., Bacanin, N., Tuba, M. and Tuba, E., *Resource scheduling in cloud computing based on a hybridized whale optimization algorithm*, Applied Sciences, vol. 9, no. 22, pp.4893, 2019.

[19] Stephanakis, I.M., Chochliouros, I.P., Caridakis, G. and Kollias, S., *A particle swarm optimization (PSO) model for scheduling nonlinear multimedia services in multicommodity fat-tree cloud networks*, In Engineering Applications of Neural Networks: 14th International Conference, EANN 2013, Halkidiki, Greece, September 13-16, 2013 Proceedings, Part II 14, pp. 257-268, Springer Berlin Heidelberg, 2013.

[20] Kakkottakath Valappil Thekkepuryil, J., Suseelan, D.P. and Keerikkattil, P.M., *An effective meta-heuristic based multi-objective hybrid optimization method for workflow scheduling in cloud computing environment*, Cluster Computing, vol. 24, pp.2367-2384, 2021.

[21] Choudhary, A., Gupta, I., Singh, V. and Jana, P.K., *A GSA based hybrid algorithm for bi-objective workflow scheduling in cloud computing*, Future Generation Computer Systems, vol.83, pp.14-26, 2018.

[22] Wu, H., Zhang, X., Song, L., Zhang, Y., Gu, L. and Zhao, X., *Wild Geese Migration Optimization Algorithm: A New Meta-Heuristic Algorithm for Solving Inverse Kinematics of Robot*, Computational Intelligence and Neuroscience, 2022.

[23] Kaur, A., Singh, P., Singh Batth, R. and Peng Lim, C., *Deep Q learning based heterogeneous earliest finish time scheduling algorithm for scientific workflows in cloud*, Software: Practice and Experience, vol. 52, no. 3, pp.689-709, 2022.

[24] Mazrekaj, A., Sheholli, A., Minarolli, D. and Freisleben, B., *The Experiential Heterogeneous Earliest Finish Time Algorithm for Task Scheduling in Clouds*, In CLOSER, pp. 371-379, May 2019.

[25] Ge, J., He, Q. and Fang, Y., *Cloud computing task scheduling strategy based on improved differential evolution algorithm* , In AIP Conference Proceedings, vol. 1834, no. 1, pp. 040038, AIP Publishing LLC, April 2017.

[26] Kuang, L. and Zhang, L., *A new task scheduling algorithm based on value and time for cloud platform*, In AIP Conference Proceedings, vol. 1864, no. 1, pp. 020017, AIP Publishing LLC, August 2017.

[27] Sun, W., Su, F. and Wang, L., *Improving deep neural networks with multi-layer maxout networks and a novel initialization method*, Neurocomputing, vol.278, pp.34-40, 2018.

[28] Dong, T., Xue, F., Xiao, C. and Zhang, J., *Workflow scheduling based on deep reinforcement learning in the cloud environment*, Journal of Ambient Intelligence and Humanized Computing, pp.1-13, 2021.

[29] A V L N Sujith, Dr. A Rama Mohan Reddy, and Dr. K Madhavi. , *Evaluating the QoS Cognizance in Composition of Cloud Services: A Systematic Literature Review* presented in International Conference on Advanced Machine Learning and Soft Computing-2018 and published in International Journal of Engineering and Technology, 7 (4.6) (2018) 141-149

[30] A V L N Sujith, Dr. A Rama Mohan Reddy,and Dr. K Madhavi., *EGCOPRAS: QoS-Aware Hybrid MCDM Model for Cloud Service Selection in Multi-Cloud Environment* Journal of Adv. Research in Dynamical & Control Systems, Vol. 11, 06, 2019

[31] A V L N Sujith, Dr. A Rama Mohan Reddy, and Dr. K Madhavi. , *QoS-Driven Optimal Multi- Cloud Service Composition Using Discrete and Fuzzy Integrated Cuckoo Search Algorithm* International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249- 8958, Volume-8 Issue- 5, June 2019.

[32] Ramesh, G., Gorantla, Venkata Ashok K and Gude, Venkataramaiah., *A hybrid methodology with learning based approach for protecting systems from DDoS attacks*, Journal of Discrete Mathematical Sciences and Cryptography, 26:5, 13171325.

Edited by: Anil Kumar Budati
Special issue on: Soft Computing and Artificial Intelligence for wire/wireless Human-Machine Interface
Received: Jan 1, 2024
Accepted: Apr 17, 2024