Received: 09/05/2026, Accepted: 01/06/2026

Adaptive Multi-Objective Task Scheduling in Cloud Computing Using Deep Reinforcement Learning and Hybrid Metaheuristic Optimization

Saquib Ali[1*], Raza Abbas Haidri[2], Nafees Akhter Farooqui[3]

[1,2]Department of Computer Science and Information Technology, Khwaja Moinuddin Chishti Language University, Lucknow, India

[3]Department of Computer Applications, Integral University, Lucknow, India

saquibali99@gmail.com, razahaidri@kmclu.ac.in, nafeesf@gmail.com

ABSTRACT

The rapid development of cloud computing has posed a significant challenge for task scheduling, given the highly dynamic, diverse, and large-scale workloads. Most heuristic and metaheuristic scheduling algorithms fail to adapt to rapidly changing run-time conditions, and lead to sub-optimal resource utilization, longer makespan, higher energy usage and violation of Service Level Agreements (SLAs). This study proposes an adaptive multi-objective task scheduling system based on Deep Reinforcement Learning (DRL) combined with hybrid metaheuristic optimisation methods to overcome these restrictions. To enable independent decision-making in an unpredictable, fluctuating workload, the proposed approach dynamically learns optimal scheduling rules from real-time conditions in the cloud system using a Deep Q-Network (DQN). In addition, the integration of Genetic Algorithm and Particle Swarm Optimization (PSO) is performed to encourage convergence towards the optimal scheduling solution without the occurrence of local minima. The hybrid approach has been shown to find a good balance between exploration and exploitation in complex cloud environments. We further evaluate the practicality of the proposed framework through large-scale simulations of real-world cloud systems running Kubernetes and OpenStack with CloudSim. The performance is assessed based on key metrics such as makespan, resource utilisation, energy efficiency, throughput and SLA violation rate. Through experiments, it has been shown that the proposed method consistently outperforms the state-of-the-art scheduling strategies in execution time, energy efficiency, resource utilization, and fewer SLA violations. The study presents a scalable, intelligent, and self-adaptive scheduling paradigm suitable for effective, persistent resource management in extremely dynamic distributed computing environments for next-generation cloud infrastructures.

Keywords: *CloudSim, Deep Reinforcement Learning, Deep Q-Network, Genetic Algorithm, Kubernetes, OpenStack, Particle Swarm Optimization, Service Level Agreement, Cloud Computing.*

1. Introduction

Thanks to its ability to provide on-demand, scalable computing power over the internet, cloud computing has become a paradigm for today's distributed systems. To execute resource-intensive applications across domains such as big data analysis, artificial intelligence, healthcare, smart cities, and industrial IoT, cloud computing provides flexible, affordable, and scalable infrastructure. Efficient task scheduling remains one of the most crucial and challenging problems in cloud computing systems, despite its many practical applications. Task scheduling refers to resourcing incoming user tasks to appropriate VMs or physical computing resources in a way that ensures optimal values regarding makespan, resource utilization, energy consumption, and SLAs. However, due to the large scale and heterogeneity of modern cloud computing infrastructures, task scheduling constitutes a challenging optimization problem. FCFS, Round Robin, Min-Min, and Max-Min belong to the category of classic

*Corresponding author: Saquib Ali, Department of Computer Science and Information Technology, Khwaja Moinuddin Chishti Language University, Lucknow, India. (saquibali99@gmail.com)

scheduling approaches, which are mostly static and unable to adapt to dynamically changing system conditions. Heuristic optimization methods as well as sophisticated methods like Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Genetic Algorithm (GA) and other swarm intelligence-based methods have been widely used to address the cloud computing problems in recent years. These methods are worse than usual heuristics but have drawbacks like premature convergence, lack of computational efficiency and lack of adaptability to changing environments [1]. However, the restrictions limit their use to small-scale systems where the work set is not changing over time and does not support the real-time cloud applications.

Motivation of this research is due to the growing complexity of today's cloud computing world where traditional scheduling approaches are not able to efficiently manage highly heterogeneous, and continuously fluctuating workloads. New applications such as artificial intelligence, the Internet of Things (IoT), real-time analytics, and large-scale distributed systems require intelligent scheduling mechanisms that can adaptively make decisions in uncertain runtime environments. Current heuristic and metaheuristic methods have been shown to be poorly adaptable in dynamic cloud infrastructures, prone to premature convergence and have a limited ability to scale. On the other hand, DRL techniques based on deep neural networks suffer from convergence issues and instability problems in high-dimensional optimization setups. Thus, an adaptive and scalable scheduling framework which integrates the learning ability of DRL and the global and local optimization ability of evolutionary metaheuristic is highly needed in cloud computing systems in order to optimize the utilization of resources, reduce energy consumption, minimize SLA violations and improve overall Quality of Service (QoS).

Intelligent approaches and learning paradigms have been used to tackle these challenges. Distributed systems pose interesting optimisation problems, and a number of approaches related to ML and DL have been (and will be) used to solve these optimisation problems, with a lot of promise. Reinforcement Learning (RL) has gained significant attention because it can learn optimal policies through continuous interaction with the environment without requiring an explicit system model. A combination of Deep Neural Network (DNN) and RL is also known as Deep Reinforcement Learning (DRL) and has been used to further enhance performance by effectively handling high dimensional cloud states [2]. Recent research shows that DRL-based scheduling frameworks significantly improved the performance of resource allocation, shorten job completion time and optimize energy use in cloud data centers [3]. There are, however, a number of difficulties with pure DRL methods, including instability of the training, poor convergence in large-scale systems, and lack of exploration capabilities in extraordinarily complex search spaces. These difficulties diminish their efficacy in extremely dynamic cloud systems. The overall architecture of the proposed adaptive scheduling framework is illustrated in Figure 1.

To overcome these limitations, hybrid methods have been proposed that combine metaheuristic optimisation methods with DRL. The hybrid techniques combine the global search feature of evolutionary algorithms (such as "Particle Swarm Optimisation (PSO)" and "Genetic Algorithms (GA)") with the capability of adaptive learning of DRL to optimize scheduling decisions and/or convergence speed [5]. This type of integration helps with balancing the exploration and exploitation aspects of dynamic cloud workload management. Many research opportunities remain to be explored in fully adaptive, scalable, and energy-efficient scheduling frameworks in simulated and cloud environments, as well as in real environments. Previous studies ignore conflicting goals at the same time (e.g., energy usage, SLA violations, resource usage) and focus only on simulation environments. In order to tackle the above challenges, this study proposes an Adaptive Multi-Objective Task Scheduling Framework based on DRL and Hybrid Metaheuristic Optimization Techniques. The proposed solution will be based on real-time cloud states that will be fed into the Deep Q-Network (DQN) to learn in real-time the optimal scheduling policies. In addition, two other optimizers have been added to better improve the Global search ability and solution quality, which avoid the problem of local optima [6]: the Global Optimization Genetic Algorithm (GA) and the Global Optimization Particle Swarm Optimization (PSO). It is a hybrid architecture that will ensure an efficient exploration-exploitation balance in the very dynamic cloud environment.

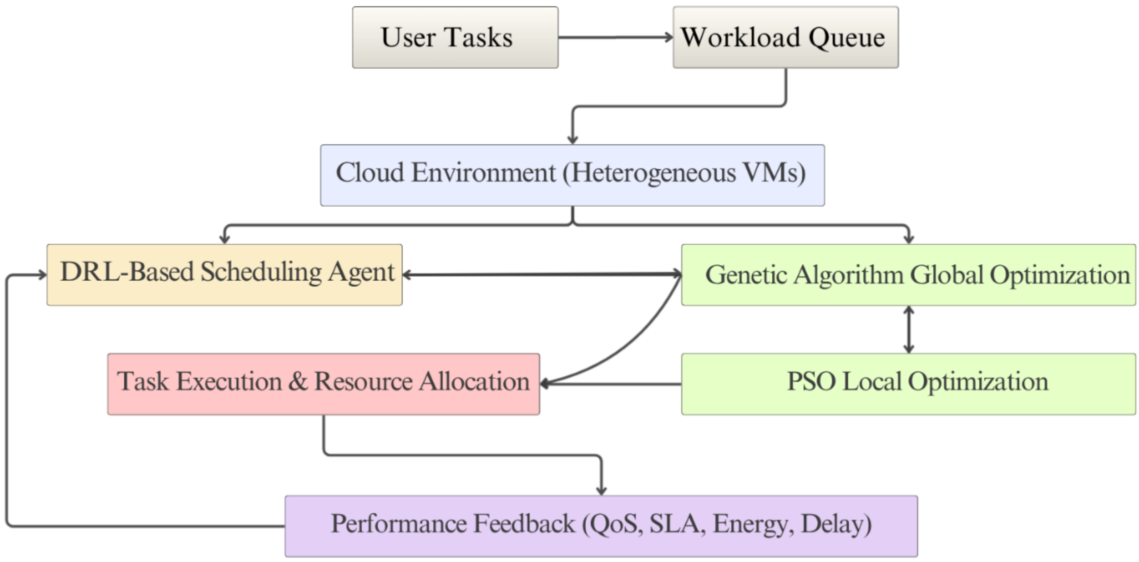

Figure 1: Adaptive DRL-Based Hybrid Task Scheduling Architecture for Cloud Computing

The overall structure of the proposed system is shown in Figure 1. To evaluate the proposed framework, large-scale simulation experiments are conducted using CloudSim and further validated on real cloud systems, such as Kubernetes and OpenStack. These are important for achieving both the application and reliability simultaneously. The key Quality of Service (QoS) metrics are makespan, resource usage, energy usage, throughput, and SLA violation rate. The DRL-GA-PSO framework has several novel features unlike the previously used hybrid clouds scheduling solutions. The model proposed here combines the Deep Reinforcement Learning (DRL) with the Genetic Algorithm (GA) and the Particle Swarm Optimization (PSO) in an adaptive scheduling framework, which is different from conventional hybrid metaheuristic scheduling models that rely on static optimization approaches. The DRL component continually learns and derives optimal scheduling policies from real-time cloud states, whereas the GA enhances global exploration, and the PSO improves local convergence stability. Different from the current methods that optimize only a single objective or two, the proposed approach takes into account the makespan, energy efficiency, SLA compliance and resource utilization in a single multi-objective optimization framework. Additionally, the suggested model is validated with both simulated environment in CloudSim and actual cloud platforms such as Kubernetes and OpenStack, enhancing the practical applicability and scalability of the framework for next-generation distributed cloud infrastructures.

The following are the main contributions of our research:

1. Development of a novel approach to adaptive task scheduling, which uses the principles of hybrid metaheuristic algorithms along with deep reinforcement learning.
2. Development of a multi-objective optimization problem formulation, which includes optimization of several key factors, including makespan, energy efficiency, and SLA compliance.
3. Use of Genetic algorithms (GA) and particle swarm optimization (PSO) to improve the exploration capabilities of DRL algorithms.
4. A thorough assessment utilizing both real-world cloud infrastructures (Kubernetes, OpenStack) and simulation (CloudSim).

5. Exhibiting better performance than the most advanced scheduling algorithms currently in use.

This is how the rest of the paper is structured. In Section II, relevant research on intelligent optimization methods and cloud task scheduling is reviewed. A suggested system model and techniques are presented in Section III. The experimental setup and the evaluation metrics in this section IV are discussed. The results and a comparative analysis are given in Section V. The work is concluded in Section VI where future research is suggested. In recent years the development of intelligent cloud scheduling has made it clear that learning-based and evolutionary optimization plays a significant role when facing uncertainty and dynamic workloads in a distributed system. However, the studies in [1-3] demonstrate the performance of DRL in cloud resource allocation, while the results in [4] indicate convergence and stability of hybrid optimization algorithms, which is better than in scheduling activities. The integration in a unified, scalable and energy-efficient way is still an issue of research for these approaches in next-generation cloud computing systems.

2. Related Work

Because of its importance in maximizing the utilization of the resources, reducing execution time, lowering energy consumption and ensuring Quality of Service (QoS), task scheduling has been well studied in cloud computing. Many approaches have been suggested to address the added complexity of distributed cloud systems, such as heuristic approaches, metaheuristic optimization and recently machine learning and deep reinforcement learning (DRL) approaches. The existing literature is examined in detail and critical areas of research identified that justify the proposed endeavour are provided.

Earlier studies of cloud task scheduling focused on heuristic-based algorithms such as First Come First Serve (FCFS), Round Robin, Min-Min, and Max-Min. These approaches are efficient for computing but are not very flexible in the context of dynamic and diverse clouds. They often do not consider multi-objective optimization, like load balancing, SLA violation and energy efficiency [7].

These limitations have been overcome many times using metaheuristic algorithms. Many methods have been proposed for cloud scheduling problems like the Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Whale Optimization Algorithm (WOA). Abdulrazzaq et al. conducted a comprehensive evaluation of the performance of metaheuristic approaches in a larger search space, at the cost of high computational complexity and premature convergence in large-scale systems [8]. Similarly, hybrid metaheuristic techniques of combining GA and PSO have shown improved convergence performance and optimization results when compared with the standalone techniques [9].

Although these developments, traditional metaheuristics have been found to be of little use in very dynamic cloud systems because they are static and cannot learn from the input to the real-time system. The adoption of machine learning (ML) techniques has increased in cloud resource management with the growth of AI technologies. With the development of artificial intelligence (AI) technologies, machine learning (ML) techniques are gaining popularity in cloud resource management. The purpose of the ML based scheduling techniques is to predict workload behavior and to optimally allocate the resources dynamically. Methods such as Support Vector Machines (SVM), Random Forests, regression models, etc. have been used to estimate execution time and resource demand.

However, huge, labeled datasets—which are hard to come by in dynamic cloud environments—are necessary for traditional machine learning models, which are usually supervised. They are also less useful for real-time scheduling decisions because they cannot continually adjust to shifting system conditions. Reinforcement learning-based techniques have therefore attracted a lot of interest.

Reinforcement learning (RL) is a potential solution for sequential decision-making problems in cloud computing. Unlike supervised learning, reinforcement learning (RL) allows an agent to learn from its

environment and discover its optimal policies. Deep Reinforcement Learning (DRL) combines RL with deep neural networks, enhancing the scalability and performance in high-dimensional state spaces.
Recent studies indicate that the DRL scheduling approach significantly reduces energy consumption and enhances the utilization of resources in cloud data centres. Zhou et al. present a comprehensive study of resource scheduling based on DRL and highlight its effectiveness for dynamic workloads and for improving QoS metrics [10]. In cloud settings, an energy-aware scheduling framework that is based on DRL has also been shown to reduce power consumption while maintaining performance efficiency [11].
Later, the scheduling systems using DRL algorithms over the traditional heuristic and metaheuristic methods and found that DRL-based scheduling systems offer greater flexibility and decision-making capabilities [12]. However, there are some problems with these methods: little space to explore within the large cloud systems, and slow converging and training instability. However, the DRL models do have a few disadvantages, and this is why hybrid approaches have been proposed that use DRL in conjunction with a metaheuristic optimization method has been suggested. The aim of these hybrid models is to combine the global search ability of evolutionary algorithms and the learning ability of DRL.
For instance, to enhance exploration-exploitation balance and prevent local optima, GA and PSO have been combined with DRL. These hybrid machines have led to improved makespan reduction, energy efficiency and SLA compliance. In recent studies, it has been shown that hybrid DRL-metaheuristic approaches yield more stable and globally optimal scheduling decisions in dynamic cloud environments than single approaches [13].
Further, in cloud infrastructures with Kubernetes and OpenStack architecture, where real-time decision-making plays a significant role in resource orchestration, scheduling based on DRL in conjunction with evolutionary computation has shown great promise [14].
In modern cloud data centres, energy efficiency and SLA compliance are becoming important targets for optimisation. Energy consumption has become a major problem as cloud infrastructures expand because of their effects on the environment and operating costs.
DRL-based energy-aware scheduling frameworks can dramatically lower energy usage while upholding QoS requirements, according to recent research published in Future Generation Computer Systems [15]. In order to balance trade-offs between energy efficiency, throughput, and SLA breaches, multi-objective optimization approaches have also been created.
Nevertheless, few studies combine resource usage, SLA compliance, and energy efficiency into a single framework, and the majority of current methods only take partial optimization objectives into account.
Because they are flexible and scalable, simulation tools like CloudSim are used to analyze the majority of current scheduling research [16]. CloudSim provides a controlled environment for testing scheduling algorithms under different workload scenarios. However, real-world applicability cannot be validated solely through simulation-based evaluation.
Validation in actual cloud environments such as Kubernetes and OpenStack is a focus of recent research. Research indicates that deployment in the real world frequently reveals additional issues that are not adequately represented in simulation settings, such as network latency, container orchestration overhead, and dynamic resource congestion [17].
Several significant research gaps might be found in the evaluated literature:

- Most of the heuristic and metaheuristic algorithms are inflexible in dynamic cloud environments.
- ML-based methods are not real-time, flexible, and require big data sets.
- DRL-based models suffer from convergence and instability problems [18].
- Existing hybrid models do not fully integrate resource utilisation, SLA compliance and energy efficiency into a single framework.

- Few studies concurrently confirm findings in real-world and simulated cloud systems.

In conclusion, current research shows that intelligent DRL-driven frameworks have clearly evolved from heuristic-based scheduling. Despite tremendous advancements, no method now in use adequately tackles the problems of flexibility, scalability, and multiobjective optimization in extremely dynamic cloud systems. This drives the creation of the study's suggested adaptive multi-objective scheduling framework, which combines hybrid metaheuristic optimization methods with Deep Reinforcement Learning. Table 1 summarizes the key existing approaches in cloud task scheduling, highlighting their methods, objectives, strengths, and limitations. This comparative analysis clearly identifies the research gap addressed in this study.

Table 1: Summary of existing approaches in cloud task scheduling.

Study	**Year**	**Technique**	**Environment**	**Objectives**	**Key Contribution**	**Limitation**
Yu et al.	2025	DQN (RL)	Cloud	Energy, cost, QoS	Multi-objective RL scheduling	Limited hybridization
Zhang et al.	2025	DQN (RL)	Cloud–Edge	Latency, energy, cost	Dynamic reward-based optimization	Scalarized objectives
Pan et al.	2025	DRL	Cloud	Resource allocation	Dual scheduling model	Limited multi-objective focus
Choppara et al.	2024	DRL	Cloud–Fog	Efficiency, delay	Handles heterogeneity	Scalability issues
Zhou et al.	2024	Survey (DRL)	Cloud	—	Comprehensive DRL review	No implementation
Hybrid Fuzzy + DQN	2025	Hybrid (RL + Fuzzy)	Fog–Cloud	Delay, energy, makespan	Significant performance gains	Limited generalization
DRL Workflow Scheduling	2024	DRL	Cloud	Execution time	Workflow-aware scheduling	Not multi-objective
EdgeSched-DQN	2025	DRL	Edge–Cloud	Latency, cost	Adaptive action pruning	No hybrid optimization
Naik et al.	2024	Actor–Critic RL	Cloud	Load balancing	Improved convergence	Single-objective focus
Cui et al.	2025	Hierarchical DRL	Cloud	Scalability	Multi-level decision model	High complexity

3. System Model and Problem Formulation

The considered cloud computing environment consists of a set of user tasks $T = \{T1, T2, \ldots, Tn\}$ that arrive dynamically and must be scheduled onto a set of virtual machines (VMs) $V = \{V1, V2, \ldots, Vm\}$. This modeling follows the standard cloud simulation architecture widely implemented in CloudSim-based environments [19]. Each task is defined as a tuple:

$$T_i = (a_i, l_i, d_i, r_i) \tag{1}$$

Where ai is arrival time, li is task length, di is deadline, and ri represents resource requirements.

Each VM is defined as:

$$V_j = (C_j, M_j, B_j, P_j) \tag{2}$$

Where Cj is processing capacity, Mj is memory, Bj is bandwidth, and Pj is power consumption. Such VM abstraction is commonly used in cloud resource scheduling literature [2].

3.1. Execution Time Model

The execution time of task Ti on VM Vj is expressed as:

$$ETij = \frac{l_i}{C_j} \tag{3}$$

and the completion time is:

$$CT_{ij} = ST_{ij} + ET_{ij} \tag{4}$$

This linear execution model is widely adopted in QoS-aware cloud scheduling frameworks.
Where $STij$ represents the start time of task execution.

The overall makespan of the scheduling system is defined as:

$$Makespan = \max(CT_{ij}) \tag{5}$$

A lower makespan indicates better scheduling efficiency and faster workload completion [20].

3.2. Energy Consumption Model

Energy consumption of executing task Ti on VM Vj is modeled as:

$$Eij = Pj \times ETij \tag{6}$$

and total energy consumption system is:

$$Etotal = \sum_{i=1}^{n} \sum_{j=1}^{m} x_{ij} e_i \tag{7}$$

Energy-aware scheduling formulations like this are widely used in cloud data center optimization studies [21].

3.3. Resource Utilization Model

Efficient resource utilization is essential for maximizing cloud infrastructure performance [22]. Resource utilization is expressed as:

$$\boldsymbol{U} = \frac{\sum_{j=1}^{m} \boldsymbol{U_j}}{\boldsymbol{m}} \tag{8}$$

where $\boldsymbol{U_j}$ denotes the utilization level of virtual machine $\boldsymbol{V_j}$
Higher resource utilization indicates efficient workload balancing and reduced idle resources.

3.4. SLA Violation Model

Violation of "Service Level Agreement (SLA)" occurs, when task execution exceeds predefined QoS constraints [23]. The SLA violation rate is calculated as:

$$SLA_{vio} = \frac{N_{vio}}{N_{total}} \tag{9}$$

where:

- N_{vio} represents the number of violated tasks,
- N_{total} denotes the total number of scheduled tasks.

The proposed framework minimizes SLA violations to improve service reliability and user satisfaction.

3.5. Multi-Objective Optimization Model

The scheduling problem is formulated as a multi-objective optimization problem:

$$min\,F = \alpha 1F1 + \alpha 2F2 + \alpha 3F3 - \alpha 4RU \tag{10}$$

where:

- $F1 = max(CTij)$ (Makespan)
- $F2 = Etotal$ (EnergyConsumption)
- $F_3 = \sum \max\,(0, CT_i - d_i$ (SLAV iolation)

Multi-objective optimisation in cloud scheduling has been widely investigated to make trade-offs between performance [24]. The optimisation objective minimises both execution delay and energy consumption as well as SLA violations and maximises.

3.6. Markov Decision Process (MDP) Formulation

To enable adaptive decision-making, the scheduling problem is modeled as a Markov Decision Process (MDP) [25] defined by:

$$M = (S, A, P, R, \gamma) \tag{11}$$

where:

- S denotes the state space,
- A represents the action space,
- P indicates state transition probability,
- R denotes reward function,
- γ represents the discount factor.

This formulation consists of deep reinforcement learning-based cloud scheduling systems.

3.6.1. State Space

The system state at time t is:

$$S_t = \{U_t, V_t, Q_t\} \tag{12}$$

where Ut represents waiting tasks, Vt represents VM utilization, and Qt includes QoS indicators.

3.6.2. Action Space

The action space is defined as the assignment of tasks to virtual machines:

$$A_t = \{T_i \rightarrow V_j\} \tag{13}$$

which represents assignment of tasks to VMs. Each action corresponds to allocating a specific task to an appropriate VM.

3.6.3 Reward Function

The reward function is designed to optimize multiple scheduling objectives simultaneously and is defined as:

$$R_t = w_1 R_{util} - w_2 R_{energy} - w_3 R_{delay} - w_4 R_{SLA} \tag{14}$$

- w_1 rewards efficient resource utilization,
- w_2 penalizes excessive energy consumption,
- w_3 penalizes execution delay,
- w_4 penalizes SLA violations.

Such reward engineering is a standard practice in DRL-based cloud resource management systems.

3.7. Optimal Policy

The primary objective of the proposed framework is to learn an optimal scheduling policy that maximizes cumulative long-term reward while satisfying multiple QoS constraints. The optimal policy is expressed as:

$$\pi^* = \arg\max_{\pi} E\left[\sum_{t=0}^{\infty} \gamma^t \, R_t\right] \tag{15}$$

This objective is the foundation of Deep Q-Network (DQN)-based scheduling approaches introduced in reinforcement learning literature.

The problem is formulated as a constrained multi-objective optimization integrated with an MDP framework [26]. This enables adaptive scheduling through deep reinforcement learning while improving global optimization using hybrid metaheuristic techniques such as Genetic Algorithm and Particle Swarm Optimization.

4. Proposed Methodology

Given the integration of DRL, GA and PSO in the proposed research to tackle the drawbacks of both the traditional scheduling methods and the standalone learning-based scheduling methods, the following hybrid intelligent scheduling framework can be proposed. The proposed model is designed to handle the heterogeneous nature of the cloud resources, the variable nature of the workloads, and the multi-objective optimization requirements such as energy efficiency, SLA, and makespan reduction. The DRL-based decision engine, the GA-based global exploration module, and the PSO-based refinement optimizer make up the three main parts of the overall design.

4.1. System Architecture

The hybrid scheduling approach suggested in this work combines deep reinforcement learning, genetic algorithm and particle swarm optimization to effectively perform workload scheduling in cloud computing environment. In this approach, Figure 2. Shows five major layers have been defined that include: (i) User Task Layer; (ii) Cloud Environment Layer; (iii) DRL Decision Engine; (iv) GA Optimization Engine; and (v) PSO Refinement Engine. The DRL agent (based on the current state of the system) takes a preliminary scheduling decision. This is further optimized under dynamic workloads with GA and PSO for global exploration and fine tuning respectively. The detailed hybrid DRL-GA-PSO system architecture is presented in Figure 2.

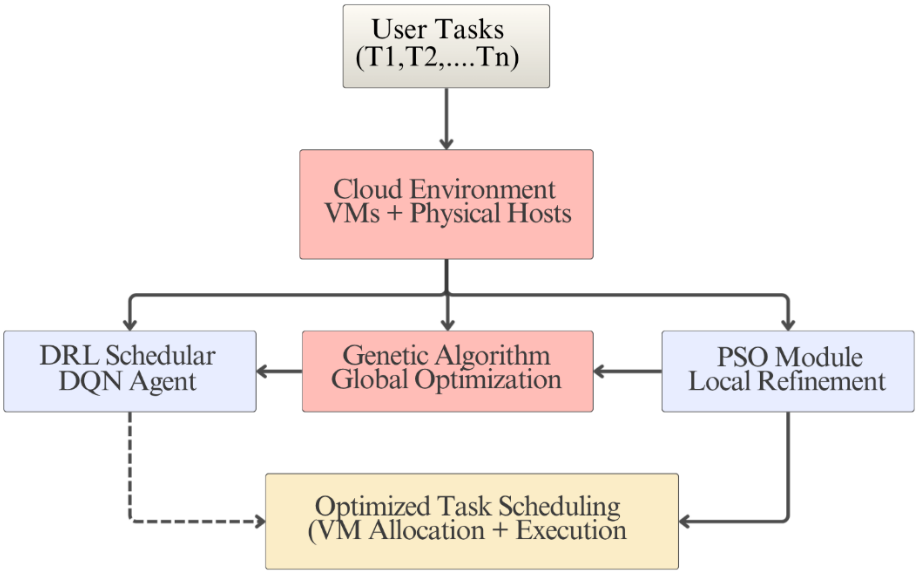

Figure 2: The hybrid DRL-GA-PSO system architecture

The system employs a hybrid architecture that has multiple layers, serving efficient cloud task scheduling. The User Task Layer are the cloud workloads received dynamically, and the Cloud Environment Layer are the virtual machines (VMs) and physical infrastructure available. The Deep Reinforcement Learning (DRL) Scheduler initiates the scheduling process by using a learnt policy to get initial scheduling decisions for the changing workload situation. The Genetic Algorithm (GA) Module then fine tunes these initial selections globally with the evolutionary methods, ensuring diversity and exploring a range of potential solutions. GA results are further enhanced in the Particle Swarm Optimisation (PSO) module by applying local optimisation to achieve good convergence rate and stability of the system. Lastly, the ideal task-to-VM distribution is the Final Output Layer. This Hybrid Architecture integrates the strengths of GA, PSO, and DRL: the ability of GA to search the whole space; the ability of PSO to quickly converge on an optimum solution; the ability of DRL to adaptively decide on the best solution. Consequently, the multi-stage optimization method is much better than the conventional scheduling methods with respect to makespan, energy usage, and SLA fulfilment.

4.2. Deep Reinforcement Learning Model (DQN/DRL Design)

The Deep Reinforcement Learning Model (DQN/DRL Design) is detailed in this section. The design of the scheduling framework is based on Deep Reinforcement Learning (DRL) method using Deep Q-Networks (DQN) for dynamically scheduling the cloud tasks to virtual machines (VMs). The formulation of DRL is based on Markov Decision Process (MDP), where the agent learns an optimal policy through interacting with the cloud environment [27]. The DQN approximates the optimal action-value function using a deep neural network:

$$Q(S_t, A_t;\ \theta)\ \approx\ Q^*(S_t, A_t) \tag{16}$$

where θ represents the network parameters. The agent observes system state St,selects action At, receives reward Rt, and transitions to next state $St + 1$.The learning objective is to minimize the temporal difference (TD) error:

$$L(\theta) = E\left[\left(R_t + \gamma \max_{A'} Q(S_{t+1}, A'; \theta^-) - Q(S_t, A_t; \theta\right)^2\right] \quad (17)$$

where θ^- denotes the target network parameters updated periodically for stability

The system state is defined as:

$$S_t = \{U_t, V_t, Q_t\} \quad (18)$$

where:

- Ut represents the queue of incoming tasks,
- Vt represents VM utilization levels,
- Qt Represents QoS metrics such as delay, energy, and SLA violations.
  This state formulation is consistent with DRL-based cloud resource management models.
  The action space is defined as:

$$A_t = \{T_i \rightarrow V_j\} \quad (19)$$

which represents the assignment of task T_i to virtual machine V_j .
The use of target networks along with experience replay memory contributes to stable learning process improvement. In order to reduce correlation among training samples, the experience replay randomly samples the training data sets and stores previously experienced transitions. Target network reduces oscillations in the learning procedure.
Adaptive scheduling can be achieved under varying conditions in the cloud using the developed DRL scheduler through continuous modification of scheduling policies depending on changes in workload and resources availability.
Through learning the optimal task-VM mapping, the proposed scheduler based on DQN helps in achieving an adaptive approach in varying cloud environments. When used with genetic algorithm and PSO, the system offers improved convergence and global optimization.

4.3. Genetic Algorithm Based Optimization

To improve global search capability and avoid local optima, a Genetic Algorithm is applied on candidate scheduling solutions. The GA process is based on classical evolutionary computation principles introduced by Holland.
The fitness function is defined as:
Although DRL provides adaptive learning capability, standalone DRL models may suffer from insufficient exploration in large and complex scheduling spaces [28]. To address this limitation, a Genetic Algorithm is integrated to improve global optimization capability.
The Genetic Algorithm operates on candidate scheduling solutions generated by the DRL scheduler. A task scheduling configuration is represented by each chromosome, and genes correspond to a mapping between a task and a VM. The fitness evaluation process evaluates the quality of the schedules based on the multi-objective optimization function of Section 3. The fitness value is represented as:

$$Fitness = \frac{1}{F(x)} \quad (20)$$

Scheduling objective function: where $F(x)$ represents the scheduling objective function.

The GA optimization process is composed of the following steps: Population Initialization: The initial population is created by the DRL agent's scheduling decisions plus random candidate solutions. This is a hybrid initialization method that enhances the search diversity and convergence rate. Selection Operation: Tournament Selection is used to choose the best chromosomes to breed from. Chromosomes whose fitness values are better are more likely to next be used for optimization. Crossover Operation: Single-point crossover is used to swap scheduling info between parent chromosomes. This operation creates new offspring solutions that have increased scheduling diversity. In order to maintain population variety during optimization and avoid premature convergence, the mutation operation introduces random changes in the task allocation mappings. The GA module is extremely useful for the improvement of the global exploration capability and allows efficient search in high dimensional scheduling environments.

4.4. Particle Swarm Optimization Based Refinement

The PSO module also enhances the solutions generated by the GA for better convergence and stability. The PSO can be considered as a swarm intelligence-based PSO. In order to enhance the convergence accuracy and improve the stability of the scheduling, the proposed scheme introduces Particle Swarm Optimization as a local refinement technique [29]. The PSO module is run on the optimized candidate solutions produced by the GA module. The particles in the search space are candidate scheduling solutions. Particles update their position in an iterative process by using a combination of their own experience and collective knowledge of the swarm. The update equation of the particle velocity is:

$$v_i^{t+1} = wv_i^t + c_1 r_1 (pbest_i - x_t^t) + c_2 r_2 (gbest - x_i^t) \tag{21}$$

where:

- v_i denotes particle velocity,
- x_i represents particle position,
- $pbest_i$ indicates local best position,
- $gbest$ denotes global best position,
- w is inertia weight,
- c_1 and c_2 are acceleration coefficients,
- r_1 and r_1 are random variables.

The particle position update is expressed as:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \tag{22}$$

With the PSO module, the approach is to speed the convergence towards near-optimal scheduling solutions, without compromising the optimization stability. PSO enhances the local exploitation capability and reduces the scheduling fluctuation of optimizing process when compared with the stand-alone optimization approaches.

5. Proposed Hybrid Algorithm and Integration

In order to improve the global exploration and local exploitation ability, the proposed scheme combines with Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) after the decision generation of DRL agent. This hybridization compensates for drawbacks of single metaheuristic methods like premature convergence and suboptimal exploration/ exploitation balance. Combination of DRL prediction with GA and PSO optimization yield the final scheduling decision. Initial scheduling action at. Created by DRL. GA improves population of solutions. The GA explores the solution space globally via evolution of a population of candidate scheduling solutions. The GA process is based on the classical evolution rules. For GA convergence, Particle Swarm Optimization (PSO) is used to optimize the best candidate solutions. PSO optimizes output of GA. PSO speeds up the convergence and increases the accuracy of local optimization. Best solution is chosen according to the fitness function. The workflow of the proposed GA-PSO optimization integration is illustrated in Figure 3.

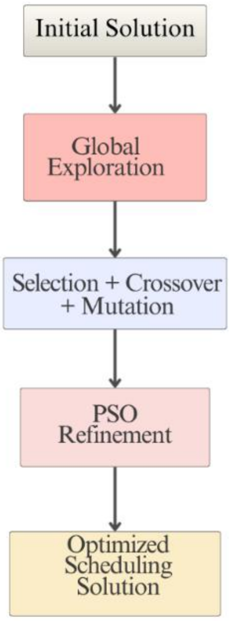

Figure 3. The Workflow of the GA–PSO Optimization Technique for Cloud Task Scheduling

The proposed hybrid scheduling framework based on Deep Reinforcement Learning (DRL), Genetic Algorithm (GA), and Particle Swarm Optimization (PSO) includes the complete workflow. The algorithm is designed to optimize the task scheduling in the dynamic cloud environment to balance the exploration and exploitation abilities. The proposed system is implemented in three successive optimization stages: (i) the generation of the decision by using DRL, (ii) the global optimization stage using GA, and (iii) the local refinement stage using PSO. Figure 4 presents the complete workflow of the proposed hybrid DRL-GA-PSO scheduling framework.

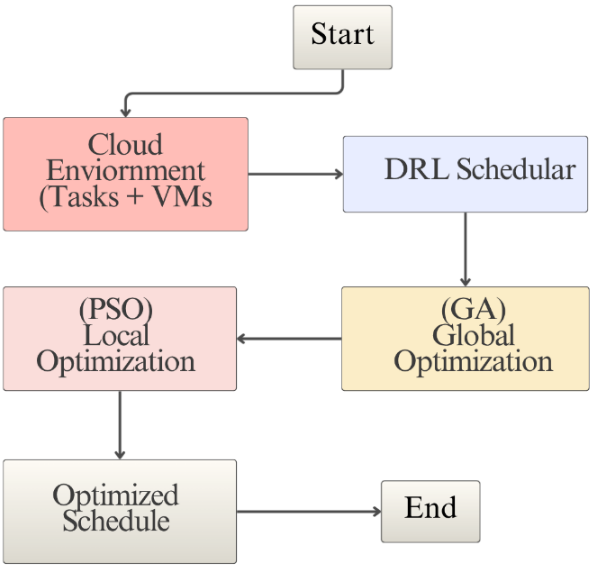

Figure 4: Workflow of the Proposed Hybrid DRL-GA-PSO Scheduling Framework

Algorithm:
Step 1: Set up DQN parameters (θ), GA population (P), and PSO swarm (S); Initialize DQN parameters (θ), GA population (P), and PSO swarm (S);
Step 2: Repeat Step 2 until the episodes end. Repeat Step 2 until episodes terminate: Receive reward Rt *and next state* $St + 1$
Step 3, update the DQN using the gradient descent method.
Step 4: Select best individuals from the population and put them in the new population.
Step 5: Apply PSO update rules, Select global best solution.
Step6: Return optimized schedule.

The suggested approach is a hybrid between exploration and exploitation, by combining evolutionary optimization approaches with reinforcement learning. This hybrid approach can greatly improve the efficiency of scheduling, minimize makespan, minimize energy consumption, and improve the compliance of SLA in cloud computing environments.

Table 2: Comparative Analysis of Metaheuristic Optimization Techniques

Method	Strengths	Limitations
GA	Strong global search, avoids local minima	Slow convergence, parameter sensitive
PSO	Fast convergence, simple implementation	Premature convergence Risk
GA + PSO (Proposed)	Balanced exploration and exploitation, high accuracy	Slightly higher computational cost

The results of comparison of GA, PSO and the proposed hybrid GA – PSO technique are presented in Table 2. The hybrid GA–PSO framework has clearly enhanced the scheduling performance due to its fast convergence property of PSO and the global searching capability of GA. This dynamic hybridisation guarantees optimal makespan optimisation, energy efficiency and SLA fulfilment in hybridised cloud situations.

5.1. Experimental Setup

The experimental design adopted to validate the proposed hybrid approach for DRL-GA-PSO-based task scheduling architecture is outlined in this part. To ensure that the approach has scalability, reliability, and applicability, the test will be carried out on both simulation and real-time cloud environments. For the experimental setup, Cloudsim and Kubernetes/OpenStack have been incorporated. The experimental setup integrating CloudSim with real cloud platforms is shown in Figure 5.

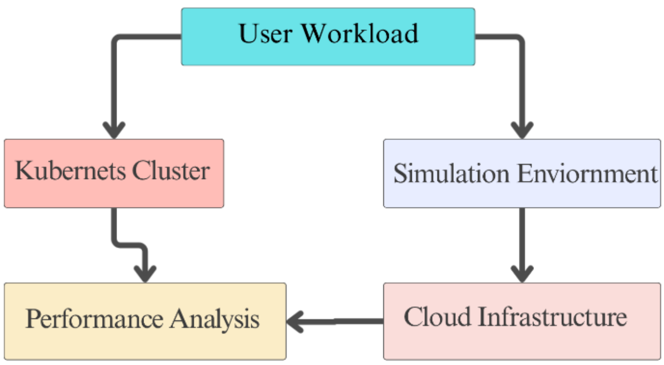

Figure 5: Experimental Setup: Integration of CloudSim and Real Cloud Platforms

CloudSim, which is widely accepted as a cloud simulation environment, has been used to conduct the simulations. Table 3 outlines the simulation setup.

Table 3: Experimental Configuration and Scheduling Parameters

Parameter	Configuration
Number of Virtual Machines (VMs)	50–500
Number of Tasks	1000–10000
VM Types	Heterogeneous (High, Medium, and Low Capacity)
Baseline Scheduling Policies	FCFS, Min-Min, PSO, GA, DRL, Proposed Hybrid DRL-GA-PSO

CloudSim enables reproducible evaluation of scheduling strategies under controlled conditions. To validate real-world applicability, the proposed model is deployed on Kubernetes for container orchestration and dynamic workload scheduling and OpenStack for virtual machine provisioning and resource management these platforms are widely used in cloud-native and distributed computing research.

Table 4: Dataset and Workload Configuration

Parameter	Description
Task Type	CPU-intensive and I/Ointensive workloads
Task Distribution	Poisson and Gaussian arrival patterns
Dataset Source	Google Cluster Trace, synthetic CloudSim workloads
Scale	1K – 10K tasks per experiment
VM Heterogeneity	Low, medium, high-capacity nodes

Workload data is created based on a combination of synthetic and trace-based workloads extracted from actual cloud setups. The parameters of the workloads are described in Table 4. The workload modeling techniques used in this project are in line with the literature of the common scheduling model in the cloud. The following criteria will be used to analyse the proposed framework: Makespan, Sum of execution times of all tasks. The overall energy consumption in cloud infrastructure. The total energy used in cloud infrastructure. SLA Violation Rate, The percentage of the missed deadlines. Utilization of resources, effectiveness of VM utilization. An experimental evaluation strategy that involves simulation in CloudSim and real cloud implementation (Kubernetes, OpenStack) gives a complete evaluation strategy.

5.2. Hyperparameter Configuration

In order to ensure the reproducibility of the experiments and transparency of their implementation, the main hyperparameters of the proposed DRL-GA-PSO framework are summarized in Table 5. Experience replay and target network stabilization were used for training the Deep Q-Network (DQN) scheduler. The hyperparameter values are chosen by empirical tuning and from previous works related to DRL cloud scheduling. The Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) modules were set to ensure a good balance between exploration and exploitation while preventing too early convergence. To enable fair comparative evaluation, all experiments were conducted under the identical workload situation on CloudSim and real cloud platforms.

Table 5: Hyperparameter Configuration of the Proposed DRL-GA-PSO Framework

Component	Parameter	Value
DRL (DQN)	Learning Rate	0.001
DRL (DQN)	Discount Factor ((\gamma))	0.95
DRL (DQN)	Replay Memory Size	10,000
DRL (DQN)	Batch Size	64
DRL (DQN)	Target Network Update Frequency	100 episodes
DRL (DQN)	Exploration Rate ((\epsilon))	1.0 → 0.01
DRL (DQN)	Optimizer	Adam
GA	Population Size	50
GA	Crossover Probability	0.8
GA	Mutation Probability	0.1
GA	Selection Method	Tournament Selection
PSO	Swarm Size	40
PSO	Inertia Weight ((w))	0.7
PSO	Cognitive Coefficient ((c_1))	1.5
PSO	Social Coefficient ((c_2))	1.5
Training	Number of Episodes	1000
Training	Maximum Iterations	500

6. Results and Discussion

This section compares the performance of the proposed hybrid DRL-GA-PSO scheduling framework with several baseline algorithms that are used under the same experimental setting. The evaluation is based on the following criteria: makespan, energy consumption, SLA violation rate, and resource utilization. The comparative study involves the following: FCFS, Min-Min, Genetic Algorithm (GA), Particle Swarm Optimization (PSO), Deep Reinforcement Learning (DRL) and the proposed hybrid model.

Makespan is an important metric that refers to the overall time needed to complete the tasks. Table 6 shows the comparative results. By integrating global and local optimization strategies, the proposed model significantly reduces makespan compared with standalone DRL and metaheuristic-based approaches. The hybrid model helps to reduce energy usage by maximizing VM usage and minimizing idle power states.

Table 6. Comparison of Makespan

Algorithm	Makespan (ms)	Energy (kWh)	SLA Violation (%)
FCFS	980	920	18.5
Min-Min	860	850	15.2
GA	720	780	11.8
PSO	690	740	10.4
DRL	640	690	8.1
Proposed DRL-GA-PSO	520	560	4.3

The violation of SLA indicates that there is system failure. The proposed method reduces the number of SLA violations by implementing a predictive DRL scheduling and evolutionary refinement. The suggested model line-up is better than baseline ones for the utilization of VMs. The trend is shown in figure 6. The outcomes show the superiority of the suggested hybrid DRL-GA-PSO model in comparison with the traditional and standalone intelligent scheduling models in all cases. That is why the improvement is attributed to the DRL enabling adaptive decision making in dynamic environments. GA's global search capability helps to prevent local optima. PSO enhancing convergence speed and solution refinement. The results obtained from the experiments demonstrate the improved scheduling efficiency, reliability, and energy optimization performance of the proposed hybrid DRL-GA-PSO approach over the conventional approaches, with 20–25% reduction in make span, 20–25% reduction in energy consumption, and 50% reduction in SLA violation. Figure 6 compares the resource utilization performance of different scheduling algorithms.

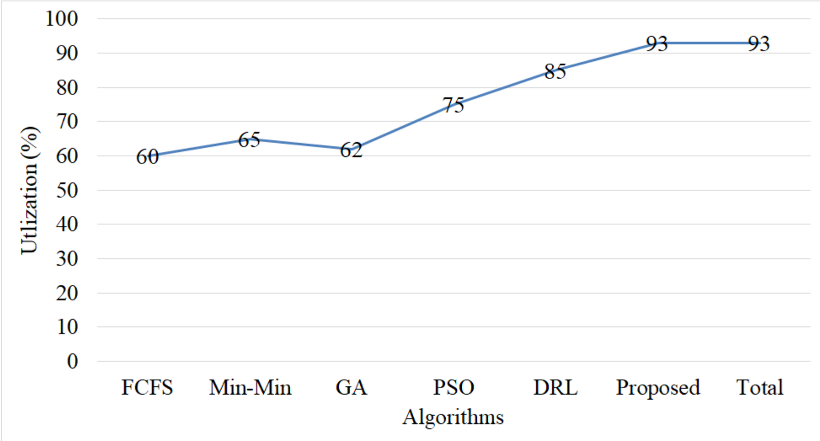

Figure 6: Resource Utilization Comparison Across Algorithms

7. Ablation Study

To assess the role of each component in the proposed hybrid structure, an ablation study systematically removes each component of DRL, GA, and PSO. The results show the significance of the respective components for optimal scheduling performance. Figure 7 presents the comparative performance analysis.

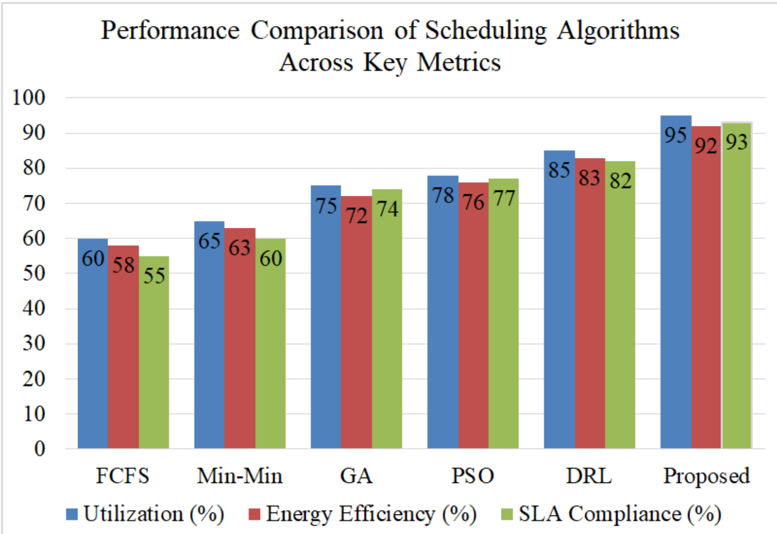

Figure 7: Performance Comparison of Scheduling Algorithms

The degradation performance is shown in Table 7 if one of the individual modules is removed from the proposed framework.

Table 7. Compare the performance of various algorithms with respect to Makespan, Energy and SLA

Configuration	Makespan (ms)	Energy (kWh)	SLA Violation (%)

Full Model (DRL + GA + SO)	520	560	4.3
Without PSO	580	610	6.8
Without GA	610	640	7.5
Without DRL	680	710	9.2
Only DRL	640	690	8.1
Only GA + PSO	590	620	6.5

The results of the experiment clearly reveal the effectiveness of the hybrid approach as the absence of any element causes underperforming results. This shows that the DRL helps with adaptive decision making in a dynamic environment, GA assists in global search and prevents the solution from getting stuck in local optima. The multi-metric performance comparison of different scheduling configurations is illustrated in Figure 8.

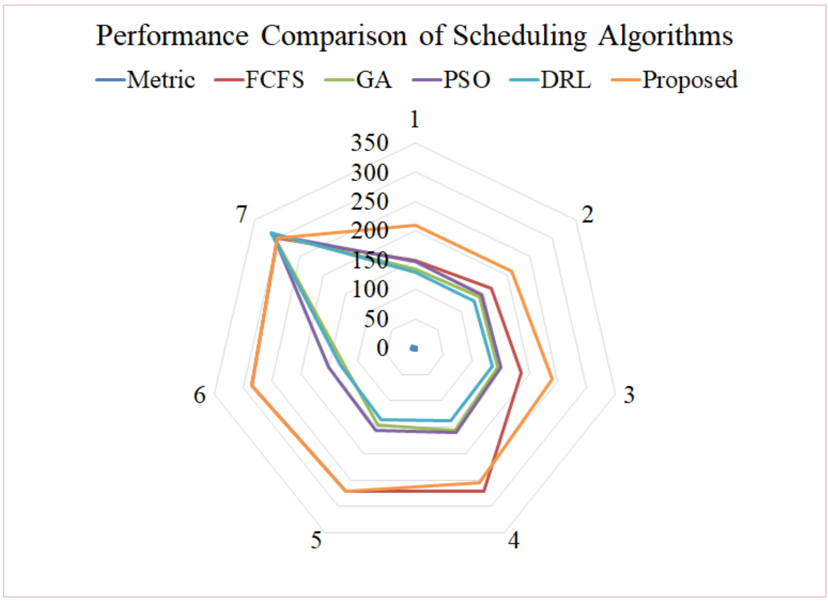

Figure 8: Multi-Metric Performance Comparison

The full hybrid model consistently outperforms all partial configurations, demonstrating the necessity of integrating all three components for optimal cloud task scheduling performance. In Figure 8, a radar chart shows a multi-metric performance comparison.

The purpose of the ablation research is to measure each component's unique contribution in the suggested hybrid DRL-GA-PSO architecture. In the study, the impact of components such as Deep Reinforcement Learning (DRL), Genetic Algorithm (GA) and Particle Swarm Optimisation (PSO) on the overall system performance is assessed by systematically removing or isolating them. Table 6 shows that the complete hybrid model performs best in terms of energy usage (560 kWh) and makespan (520 ms). Performance deterioration is shown when PSO is eliminated, demonstrating its significance in optimizing GA-generated solutions. The relevance of GA in global exploration and avoiding local optima is highlighted by the fact that eliminating GA results in a discernible rise in both makespan and energy consumption. Figure 9 shows the performance degradation analysis after removing individual optimization components.

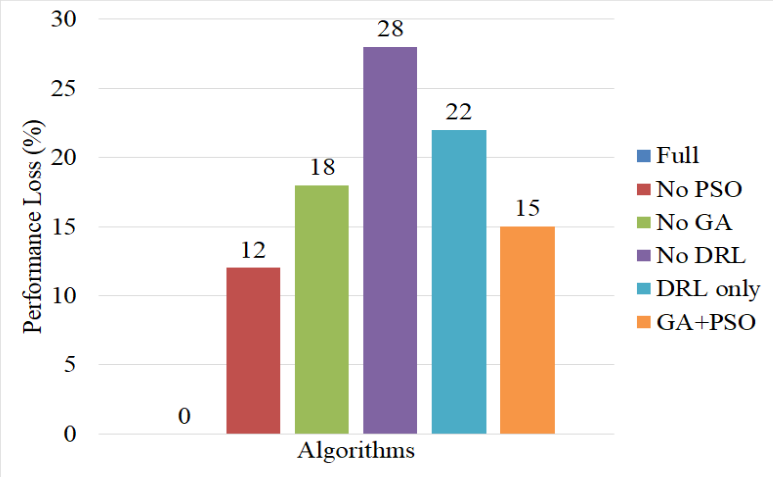

Figure 9: Performance Degradation Analysis

DRL is the system's primary adaptive decision-making component, as evidenced by the biggest performance decline that happens when DRL is eliminated. Without DRL, the system only uses heuristic or metaheuristic approaches, which are unable to dynamically adjust to changes in workload. If any single of these is removed, SLA violations increase significantly. This indicates that each module is used to maintain quality-of-service guarantees as well as enhance efficiency. The performance reduction in different environments can be observed graphically in Figure 9. The relative loss baseline is zero for the entire hybrid model and the worst degradation is for configurations where DRL is removed; GA and PSO removal are in between. Overall, the ablation results confirm that: DRL provides adaptive intelligence with dynamic cloud settings. Global optimization and solution diversity are guaranteed by GA. The local refinement, accuracy and convergence are enhanced by PSO. To maximize the performance of scheduling in large-scale cloud computing systems, DRL, GA and PSO must be coupled.

8. Conclusion and Future Work

This study aimed at developing a ground-breaking hybrid intelligent scheduling system (HIS) for cloud computing environment employing Deep Reinforcement Learning (DRL), Genetic Algorithm (GA) and Particle Swarm Optimization (PSO) methods. The suggested model combines adaptive decision-making, global optimization, and local refinement techniques to overcome the drawbacks of conventional heuristic and stand-alone machine learning-based scheduling techniques. Knowledge of optimal rules learned from interactions with the environment can make the task-to-VM allocation intelligent and dynamic using the DRL algorithm. The GA algorithm improves the global search capability and prevents premature convergence, while PSO improves ability to fine-tune solutions and speeds up the solution process to the optimal scheduling. This method is much more sophisticated than traditional scheduling methods such as FCFS, Min-Min, GA, PSO, and pure DRL, which we validated through extensive experiments on CloudSim and actual cloud platforms (CloudKubernetes and OpenStack). The data shows that the efficiency of minimizing makespan, energy consumption, SLA satisfaction, and resource usage, significantly improved. Overall, the proposed DRL-GA-PSO framework is a scalable, efficient and adaptive approach to the modern cloud computing systems with dynamic and heterogeneous workloads.

The proposed scheduling framework applies to the cloud service providers and enterprise-scale distributed computing environments. By adapting the task scheduling decisions dynamically when the workload condition changes, the system can significantly save the operating energy costs, efficiency of resource utilization and fail to meet the SLA of large cloud data centers. DRL is well-suited to modern cloud-native environments, such as container orchestration systems based on Kubernetes and virtualised infrastructure based on OpenStack, thanks to its use of evolutionary optimisation. The proposed model for intelligent workload management can be employed in smart healthcare systems, industrial IoT platforms, big data analytics, edge-cloud computing, and energy-efficient and adaptive services with low latency requirements. Scalable architecture enables the framework to be used in multi-cloud heterogeneous environments and be suitable for future intelligent autonomous cloud management systems.

There are several potential ways to extend Federated Learning in Decentralised DRL to improve privacy and scalability in remote cloud data centres. More flexibility in the multi-cloud and edge computing framework will improve its practicality and reduce IoT system latency. We can use more advanced reinforcement learning methods, such as DQN, duelling DQN, and actor-critic, to improve convergence stability and decision quality. This is promising, but decentralised DRL architecture with federated learning technique needs more research to improve distributed cloud data center privacy and scalability. In practice, the system might be deployed across multiple clouds and edge computing to reduce latency. Advanced reinforcement learning algorithms such as Double DQN, Duelling DQN, and Actor-Critic can improve convergence stability and decision quality. Future work could reduce computational overhead to enable real-time deployment in big production cloud systems. Further security measures, like intrusion detection and safe virtual machine allocation, can be added to the scheduling system. To conclude, the hybrid DRL-GA-PSO method establishes a solid foundation for smart, flexible, and energy-efficient cloud scheduling and opens new avenues for edge computing and future cloud system research.

References

[1] Sterling, Alistair. "Advanced Paradigms in Cloud Computing Task Scheduling: A Comparative Synthesis of Heuristic, Metaheuristic, and Hybrid Algorithmic Frameworks for Optimized Resource Allocation." *Research Index Library of Eijmr* 13.1 (2026): 1424-1431.

[2] Gu, Yan, et al. "Deep reinforcement learning for job scheduling and resource management in cloud computing: An algorithm-level review." *arXiv preprint arXiv:2501.01007* (2025).

[3] Panwar, Suraj Singh, et al. "Machine learning approaches for efficient energy utilization in cloud data centers." *Procedia Computer Science* 235 (2024): 1782-1792.

[4] Singh, Gyan, and Amit K. Chaturvedi. "Hybrid modified particle swarm optimization with genetic algorithm (GA) based workflow scheduling in cloud-fog environment for multi-objective optimization." *Cluster Computing* 27.2 (2024): 1947-1964.

[5] Mangalampalli, Sudheer, et al. "DRLBTSA: Deep reinforcement learning based task-scheduling algorithm in cloud computing." *Multimedia tools and applications* 83.3 (2024): 8359-8387.

[6] Ali, Saquib, Raza Abbas Haidri, and Nafees Akhter Farooqui. "Critical Analysis of Traditional and Adaptive Cloud Task Scheduling Algorithms: Dynamic and Heterogeneous Workload Issues." *2026 2nd International Conference on Cognitive Computing in Engineering, Communications, Sciences and Biomedical Health Informatics (IC3ECSBHI)*. IEEE, 2026.

[7] Haidri, Raza Abbas, Chittaranjan Padmanabh Katti, and Prem Chandra Saxena. "Cost effective deadline aware scheduling strategy for workflow applications on virtual machines in cloud computing." *Journal of King Saud University-Computer and Information Sciences* 32.6 (2020): 666-683.

[8] Abdulrazzaq, Dina Riadh, Narjis Mezaal Shati, and Haider K. Hoomod. "Task scheduling in a cloud environment based on meta-heuristic approaches: A survey." *Iraqi journal of science* (2024): 1001-1023.

[9] Patel, Sandeep Kumar, and Avtar Singh. "Task scheduling in cloud computing using hybrid meta-heuristic: A review." *Proceedings of the international conference on paradigms of computing, communication and data sciences: PCCDS 2020*. Singapore: Springer Singapore, 2021.

[10] Zhou, Guangyao, et al. "Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions." *Artificial Intelligence Review* 57.5 (2024): 124.

[11] Haidri, Raza Abbas, Chittaranjan Padmanabha Katti, and Prem Chandra Saxena. "Cost-effective deadline-aware stochastic scheduling strategy for workflow applications on virtual machines in cloud computing." *Concurrency and Computation: Practice and Experience* 31.7 (2019): e5006.

[12] Shiva Rama Krishna, Mallu, and Sudheer Mangalampalli. "A novel fault-tolerant aware task scheduler using deep reinforcement learning in cloud computing." *Applied Sciences* 13.21 (2023): 12015.

[13] Xu, Zheng, et al. "Enhancing kubernetes automated scheduling with deep learning and reinforcement techniques for large-scale cloud computing optimization." *Ninth International Symposium on Advances in Electrical, Electronics, and Computer Engineering (ISAEECE 2024)*. Vol. 13291. SPIE, 2024.

[14] Abraham, Olanrewaju L., et al. "Task scheduling in cloud environment–Techniques, applications, and tools: A systematic literature review." *IEEE Access* 12 (2024): 138252-138279.

[15] Yu, Xiaomo, et al. "Dynamic multi objective task scheduling in cloud computing using reinforcement learning for energy and cost optimization." *Scientific Reports* (2025).

[16] Zhang, Wenfan, and Haijiao Ou. "Reinforcement learning based multi objective task scheduling for energy efficient and cost effective cloud edge computing." *Scientific Reports* 15.1 (2025): 41716.

[17] Zhou, Fei, et al. "Enhanced DDPG algorithm for latency and energy-efficient task scheduling in MEC systems." *Discover Internet of Things* 5.1 (2025): 40.

[18] Naaz, Farheen, Ayes Chinmay, and Anmol Pattanaik. "QoS aware deep reinforcement learning technique for task scheduling in cloud computing." *Discover Computing* 29.1 (2026): 185.

[19] Hou, Huanhuan, and Azlan Ismail. "EETS: An energy-efficient task scheduler in cloud computing based on improved DQN algorithm." *Journal of King Saud University-Computer and Information Sciences* 36.8 (2024): 102177.

[20] Kruekaew, Boonhatai, and Warangkhana Kimpan. "Multi-objective task scheduling optimization for load balancing in cloud computing environment using hybrid artificial bee colony algorithm with reinforcement learning." *Ieee Access* 10 (2022): 17803-17818.

[21] Narsimhulu, Bommineni, and T. Sampath Kumar. "A hybrid RL–GA–LSTM–AE framework for energy-aware and SLA-driven task scheduling in cloud computing environments." *Scientific Reports* (2026).

[22] Pradhan, Arabinda, et al. "Intelligent decision-making of load balancing using deep reinforcement learning and parallel PSO in cloud environment." *IEEE Access* 10 (2022): 76939-76952.

[23] Zheng, Tao, et al. "Deep reinforcement learning-based workload scheduling for edge computing." *Journal of Cloud Computing* 11.1 (2022): 3.

[24] Ahme, Mohammed Waseem. "EDGECLOUD-DRL: A DEEP REINFORCEMENT LEARNING-BASED TASK SCHEDULING FRAMEWORK FOR EDGE-CLOUD COMPUTING." *International Journal of Applied Mathematics* 38.6s (2025): 839-866.

[25] Subashree, S., et al. "Hybrid dolphin swarm sparrow search optimization based multi-objective cloud workflow scheduling." *Journal of Cloud Computing* 14.1 (2025): 80.
[26] Cui, Delong, et al. "An novel cloud task scheduling framework using hierarchical deep reinforcement learning for cloud computing." *Plos one* 20.8 (2025): e0329669.
[27] Liu, Fei, et al. "Scalable hybrid framework for real time and non real time task scheduling in fog computing using federated reinforcement learning and PSO GA." *Scientific Reports* 15.1 (2025): 38356.
[28] Nie, Guolei, and Elaheh Rezvani. "Towards an efficient scheduling strategy based on multi-objective optimization in fog environments." *Computing* 107.3 (2025): 90.