International Journal of Applied Mathematics

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

DEEP REINFORCEMENT LEARNING FOR SLA-AWARE CLOUD RESOURCE

Govind Venugopal[1*] , Prithvi Kumar Badiga[2]

Arizona State University

Abstract:

In cloud computing, a fast-changing environment, compliance with Service Level Agreement (SLA) alongside optimal use of resources is a continuing multi-faceted challenge. Rule-based or otherwise heuristic driven traditional resource allocation strategies are not flexible enough, nor do they possess the predictive capabilities needed in an ever-changing multi-tenant cloud environment. The work introduces a brand new framework which uses Deep Reinforcement Learning (DRL) to accomplish SLA-aware provisioning and management of resources in clouds. This formulation uses the cloud environment as a Markov Decision Process (MDP) with an agent learning the best policy to allocate the resources by interacting repeatedly with dynamic cloud workloads, user demands and SLA commitments.It combines advanced Deep Reinforcement Learning (DRL) architectures (e.g., Deep Q-Networks (DQN) or Proximal Policy Optimization (PPO)) and dynamically allocates computing resources in a fashion that balances performance guarantees and cost-effectiveness trade-offs. One of the most significant contributions of this paper is the in-corporation of SLA parameters into the reward function that allows the learning agent to internalize and prioritize the SLA driven performance metrics of latency, throughput, and availability. Extensive experimental testing of real-world cloud data sets and emulated system behaviors proves the ability of the model to minimize SLA breaches, decrease resource idle time, and excel the baseline algorithms in adaptive elasticity.The analyses also show that, by optimizing resource providers and decision making, the model promotes its scalable learning-based approach to next-generation cloud resource management and covers the pathway towards more robust responsive and economically dependable cloud services.

Keywords Deep Reinforcement Learning (DRL),SLA-Aware Resource Management,Cloud Computing,QoS-Aware Scheduling, Service Level Agreement (SLA)

1. INTRODUCTION

1.1 Background and Motivation

Cloud computing has transformed the way IT services are to be delivered giving individuals and enterprises alike on-demand, scalable, and cost-effective infrastructure. With the increase in demand of the cloud service, the very idea of meeting the Service Level Agreement (SLA)

Received: August 13, 2025

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

requirement in an efficient and competent manner of managing the cloud resources has emerged as a prime issue in the minds of the cloud service providers (CSPs). SLAs form part of the contractual agreements between clients and service providers and stipulate the level of performances expected by the clients in terms of the uptime, latency, throughput and availability. The occurrences of non-compliance usually lead to penalties, loss of revenue, and damaged reputations.Resource scheduling and provisioning algorithms traditionally have recourse to some heuristic or metaheuristic method like the Particle Swarm Optimization (PSO) [2], Genetic Algorithms (GA) [3], and priority queues, in cloud environments. Such approaches are only partially satisfactory, and do not cope well with dynamic, unpredictable workloads and changing service needs, under stringent SLA requirements. Also, the manual adjustment process of parameters, inability to tune to the real-time variation and lack of generalization in different contexts restrict their applicability to contemporary, heterogeneous cloud systems.With Deep Reinforcement Learning (DRL), there lies a potential breakthrough in dynamic understanding, autonomous and adaptive cloud resource management. Because they can learn the effect on experience and adapt resource provisioning policies on a real-time basis, DRL agents are suited to create the complex trade-offs involving resource cost, performance efficiency and SLA servicing [1][4][5].

1.2 Problem Statement

The fundamental issue that this study has tried to solve is the dynamic scheduling and provisioning of the cloud resources and satisfying the SLA with the help of DRL-based approaches. The unsteadiness of workloads, and the cloud resource heterogeneity demand the sort of a solution that not alone is fast in responding to the changing demands, but also a solution that learns the best policies to be made. Current DRL uses in cloud computing have had commendable success in provisioning resources [5][9] but few take an explicit approach of including SLA parameters in the learning framework. In its absence, the system would likely prioritise throughput or energy-efficiency at the cost of contractual performance requirements.

Thus, it is paramount to provide SLA-sensitive DRL framework, in which it is possible to directly consider SLA constraints as the part of learning goals and encourage the agent to prioritize SLA-specific tasks, punish the Beurocratic violation, and consider resource consumption as a whole.

Received: August 13, 2025

International Journal of Applied Mathematics

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

1.3 Research Objectives

This research is driven by the following objectives:

- To design a **deep reinforcement learning-based architecture** for intelligent resource management in cloud computing environments.
- To integrate **SLA-awareness** directly into the DRL reward mechanism, ensuring that latency, availability, and throughput are optimized under contractual bounds.
- To evaluate the proposed framework against traditional and DRL-based benchmarks using **real-world datasets** and **simulated cloud platforms**.
- To propose a generalizable model capable of adapting to diverse cloud environments, including edge, fog, and hybrid computing infrastructures.

1.4 Literature Context and Gap Identification

Earlier research in job scheduling and resource provisioning has enhanced the cloud performance in a rampant way. Wei et al. [1] offered DRL-Scheduling, a smart framework to do the QoS-aware job scheduling in clouds. As their work showed, DRL may rule the roost over conventional heuristics in rich settings. Conversely, Wang et al. [4] implemented multi-agent DRL to multi-objective workflow scheduling and effectively addressed the cooperation of the agents in distributed systems.Cheng et al. [5] proposed a model named DRL-Cloud designed based on DRL to provide task scheduling and resources provisioning with SLA partially enforced. Kardani-Moghaddam et al. [6] provided ADRL, which is a hybrid anomaly-aware DRL model of resources scaling; however it handled anomalies in the workload still absent the incorporation of granular aspects of SLA. Rjoub et al. [7] and Shao et al. [8], instead, explored trust and GPU scheduling, respectively, and are, thus, part of the widening of scheduling research territory, but they do not directly address SLA sensitivity.The work of Tong et al. [9], in its turn, presented a framework to schedule based on Deep Q-Learning and demonstrated a strong level of adaptation of the model; however, the reward function used by the model prioritized the successful completion of the tasks at the expense of SLA compliance. Similarly, Guo et al. [13] have integrated imitation learning learning in the cloud scheduling, which augments the decision making, though does not meet SLA goals.Other approaches, like resource allocation based on a double auction, introduced by Li et al. [12] and meta reinforcement learning approaches [11] specialize in pricing and broad applicability but lack

Received: August 13, 2025

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

real-time cost of SLA violations. Karthiban and Raj [15] offered fast green computing model based on modified DRL, with priority given to the energy efficiency rather than SLA conditions. Thus, there still exists an enormous gap in DRL research applied especially to SLA-considering decisions in large dynamic cloud systems.

1.5 SLA-Awareness Together With Cloud Environments Significance

Severe SLAs control cloud service models which is composed of Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS). Such agreements determine QoS values to be maintained by the providers. A system unaware of the SLA requirements may lead to degraded services, unsatisfied users and economical fines.The current DRL implementations tend to optimize a more general concept of utility without actually punishing the policy violations, which is the policy violation. By providing SLA parameters into the reward, it is possible to do SLA-centric policy learning, thereby providing intelligent trade-offs between policies on cost and performance. The study therefore investigates reward shaping method and policy limitation, which penalize punishment of SLA violation in the decision making process of the DRL agent.

1.6 Challenges in SLA-Aware Resource Management

Designing an SLA-aware DRL system presents several technical and theoretical challenges:

Reward Function Design: Balancing immediate performance metrics with long-term SLA satisfaction is complex. The system must penalize SLA breaches while encouraging resource efficiency [14][30].

State Space Explosion: Modeling a dynamic cloud system with numerous VMs, containers, and application dependencies leads to high-dimensional state representations [21][25].

Delayed Rewards: SLA violations are often detected post-execution, making it difficult for the agent to correlate actions with outcomes [24].

Scalability and Generalization: Solutions must scale across heterogeneous data centers, supporting multiple tenants and diverse workloads [10][28].

Adaptability to Real-time Workloads: The system must respond promptly to sudden changes in resource demand, ensuring uninterrupted service delivery [20][26].

Hybridization with Other Learning Models: Combining DRL with supervised learning, imitation learning [13], or anomaly detection mechanisms [6] introduces architectural and computational complexity.

Received: August 13, 2025

International Journal of Applied Mathematics

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

1.7 State-of-the-Art Methods

Recent works have tackled various elements of DRL in cloud environments:

Dynamic task offloading and resource allocation in edge and mobile cloud systems have been successfully implemented using **deep Q-networks** and **actor-critic methods** [17][20].

Energy-aware scheduling using DRL has gained traction, especially for green data centers and 5G-enabled H-CRAN systems [22][23].

Cost-aware scheduling frameworks, like the one proposed by Cheng et al. [30], illustrate the benefits of DRL in minimizing operational costs, albeit without strict SLA models.Some researchers, like Sun et al. [24], proposed **anchor graph hashing** integrated with RL for cell activation, showcasing advanced DRL integration with domain-specific knowledge.

Others explored **multi-objective DRL approaches** combining SLA, cost, and energy efficiency [19][27], demonstrating the viability of simultaneous optimization in a unified model.

However, **no unified framework currently exists** that holistically incorporates SLA compliance, workload prediction, resource cost, and application QoS into a deep reinforcement learning architecture optimized for cloud infrastructure.

1.8 Research Contributions

This research contributes the following:

1. A **SLA-sensitive DRL framework** for adaptive cloud resource provisioning.

2. A **custom reward shaping methodology** that penalizes SLA violations and promotes compliance-driven decision-making.

3. Use of **deep neural networks and advanced DRL algorithms** (e.g., PPO, DDPG, DQN) to manage high-dimensional states and dynamic policies.

4. Extensive **performance evaluation** using real-world cloud datasets and simulation platforms to validate the model's effectiveness against traditional scheduling approaches and state-of-the-art DRL baselines.

1.9 Structure of the Paper

The rest of the paper is organized as follows:Section 2 reviews the **related works** in greater detail, covering heuristic, metaheuristic, and reinforcement learning-based cloud resource scheduling techniques.Section 3 presents the **proposed SLA-aware DRL framework**, including architectural design, reward function, and learning algorithm.Section 4 details the **experimental setup**, including

Received: August 13, 2025

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

datasets, simulation environments, and evaluation metrics.Section 5 discusses the **results**, comparing the proposed model to benchmark methods.Section 6 concludes with **key findings**, limitations, and future research directions.

2. RELATED WORKS

2.1 Classical Scheduling Techniques in Cloud Computing

Cloud resource scheduling has traditionally utilized heuristic and metaheuristic algorithms to reduce task placement and resource usage. Particle Swarm Optimization (PSO), suggested by Ebadifard and Babamir [2], enhanced task scheduling with load balancing as its emphasis. In the same vein, Keshanchi et al. [3] improved Genetic Algorithms through the use of priority queues for better cloud scheduling. While effective in static or semi-dynamic conditions, these methods suffer from scalability limitations and lack adaptability in real-time, SLA-sensitive scenarios.Moreover, evolutionary and swarm-based algorithms such as Firefly optimization [28] and Whale Optimization Algorithm (WOA) [29] have been applied to multi-objective task scheduling problems. However, these models generally lack the ability to learn from evolving patterns in workload distribution, and they operate under predefined fitness functions, which restrict their flexibility.

2.2 Appearance of Reinforcement Learning in Cloud Environments

The advent of Reinforcement Learning (RL) was a departure from static optimization towards learning-based adaptive methods. Wei et al. [1] proposed DRL-Scheduling, a QoS-aware job scheduling system, to illustrate how DRL agents can surpass rule-based techniques in changing workloads. Cheng et al. [5] further expanded this idea with DRL-Cloud, where DRL was used for both resource provisioning and scheduling. Kardani-Moghaddam et al. [6] introduced ADRL, a hybrid model that combined anomaly detection with DRL for proactive resource scaling. Their approach helped in mitigating SLA violations by predicting anomalies ahead of time. Yet, the model still did not incorporate SLA parameters into the learning process of DRL. Tong et al. [9] utilized Deep Q-Learning for scheduling jobs, confirming the efficacy of DRL for acquiring the optimal scheduling policies. However, the reward functions in those methods usually favored throughput and latency without directly embedding SLA penalties or service guarantees.

2.3 SLA-Aware Optimization: Current Gaps and Attempts

Although SLA-awareness is paramount in cloud, few models have addressed it explicitly. Cheng et al. [30] proposed a cost-conscious scheduling model based on DRL, where the cost for each job was minimized, but SLA terms were assumed implicitly. Guo et al. [13] integrated DRL with imitation

Received: August 13, 2025

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

learning to accelerate and improve decision-making in resource scheduling, but not so much to maintain SLA compliance.

Li et al. [12] investigated resource allocation with a double-auction mechanism, motivated by reinforcement and belief learning. Although their model considered pricing and availability, its SLA-awareness was modest. In the same manner, Wang et al. [4] also proposed a multi-agent DRL method for scheduling workflows, although the SLA aspect was loosely integrated with reward mechanisms.

2.4 Edge and Mobile Cloud Computing using DRL

As edge and mobile cloud paradigms are being extended, task offloading and distributed scheduling have gained significant prominence. Wang et al. [11] introduced a meta-reinforcement learning paradigm for rapid adaptive task offloading in edge computing and experienced immense improvement in responsiveness. Zhang et al. [22] presented an energy-aware allocation mechanism for H-CRANs. Sun et al. [24] presented an anchor graph hashing-boosted RL for cloud RAN auto-cell activation with distributed resource management in mind. In the same vein, Shan et al. [26] introduced "DRL+" for smart resource allocation in mobile edge computing. Although these models provide speed and robustness, straightforward SLA modeling is an ancillary issue.

2.5 Multi-Objective and Trust-Aware Scheduling

Scheduling in cloud settings is usually with contradicting goals like cost, latency, and energy. Wang et al. [18] introduced a DRL-based algorithm for energy-efficient resource allocation in Cloud-RANs considering mainly energy. Rjoub et al. [7] proposed Big Trust Scheduling, combining trust parameters into task scheduling and enhancing security and accountability but not actually defining SLA terms. Ismayilov and Topcuoglu [21] utilized neural networks within a multi-objective evolutionary algorithm for dynamic workflow scheduling. This combination enabled SLA-like goals to be achieved under some constraints but did not include end-to-end contractual modeling.

2.6 Deep Learning Variants and Hybrid Models Integration

More and more works are integrating DRL with other paradigms of learning. Feng et al. [14] used cooperative DRL to allocate resources in blockchain-based mobile-edge computing. Karthiban and Raj [15] proposed a modified DRL algorithm to enable fair resource allocation and green computing. Lu et al. [17] optimized task offloading strategies with light DRL, providing computational efficiency but no SLA-constrained resource management.Dong et al. [16] discussed DRL in the context of cloud manufacturing, prioritizing manufacturing task deadlines but not universal SLA metrics. Juarez et al. [23] presented energy-aware scheduling for parallel cloud tasks, introducing another level of optimization to the DRL framework.

Received: August 13, 2025

International Journal of Applied Mathematics

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

2.7. Motivation for the Present Study

This work is poised to fill the glaring gap between DRL-based resource scheduling of cloud resources and SLA ful fill ment. By integrating SLA measures directly into the DRL reward mechanism, the developed model seeks to establish a balanced, adaptive, and scalable resource provisioning mechanism. As distinct from the existing methods, this research emphasizes dynamic adaptability to SLA change in real time, decision spaces of high dimensionality, and cheaply scalable solutions—hence providing a more contractual-aware and holistic optimization approach for contemporary cloud infrastructures.This work is also very novel as it introduces an end-to-end SLA-aware deep reinforcement model, which can dynamically learn to adapt over multiple service dimensions while inflicting minimal SLA violations on dynamic and heterogeneous cloud environments.

PROPOSED SYSTEM BLOCK DIAGRAM:

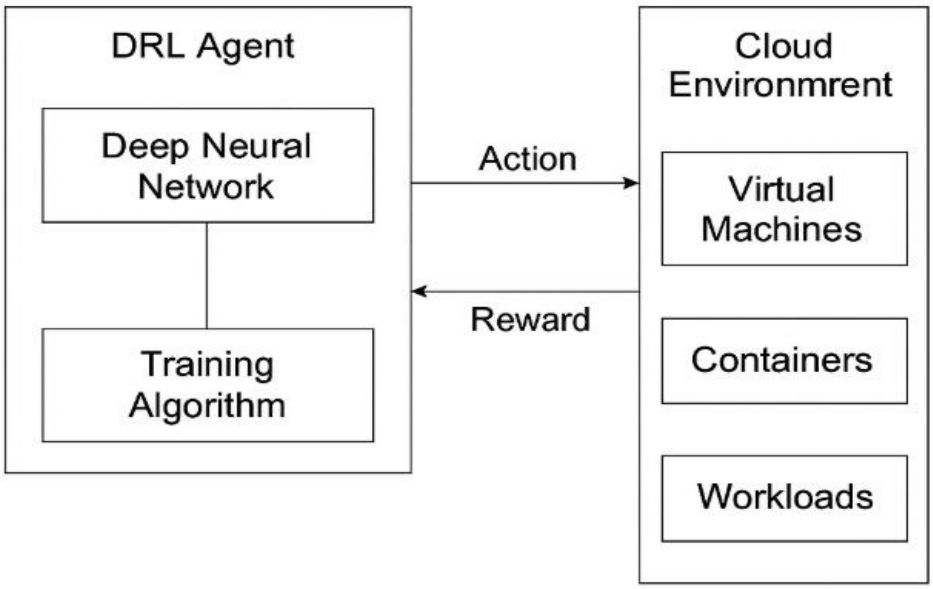

Figure 1: Proposed Block Diagram

3. PROPOSED METHODOLOGIES

3.1 Overview of the Proposed Framework

The proposed work introduces an **end-to-end Deep Reinforcement Learning (DRL) framework** tailored specifically for **SLA-aware resource management in dynamic cloud computing environments**. Unlike conventional optimization and existing DRL-based models that primarily focus on throughput or energy efficiency, this research integrates **explicit SLA parameters**—such as latency,

Received: August 13, 2025

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

availability, execution deadline, and throughput guarantees—into both the **state representation and the reward function**.

The architecture comprises three major components:

1. **Cloud Environment Simulator** (with SLA constraints)
2. **DRL Agent** (using Deep Q-Network and PPO variants)
3. **Policy Engine with SLA-Aware Reward Optimizer**

This framework ensures that **task allocation**, **resource provisioning**, and **auto-scaling** are dynamically managed while respecting pre-agreed SLA thresholds.

3.2 Cloud Environment Modeling

The simulated cloud environment models a **multi-tenant IaaS cloud** that includes:

- Heterogeneous Virtual Machines (VMs)
- Containerized workloads
- SLA descriptors associated with each workload

Each task arriving at the cloud system carries a tuple:

<Task_ID, CPU_req, Memory_req, Deadline, Priority, SLA_penalty>

The cloud system also includes a real-time SLA-monitoring subsystem to observe:

- Response time
- Task drop rate
- Availability metrics

These are used not only for monitoring but also as feedback to the DRL agent during training and inference.

3.3 DRL Formulation

The problem is modeled as a **Markov Decision Process (MDP)**, defined as follows:

State Space (S):

Includes current VM states, workload queue lengths, resource utilization metrics, SLA counters, and deadline buffers.

Action Space (A):

Received: August 13, 2025

International Journal of Applied Mathematics

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- Allocate a task to a VM
- Migrate workload
- Scale up/down resource instance
- Reject/Defer task (with SLA penalty)

3.4 DRL Algorithm Selection and Customization

This work leverages two DRL variants for experimentation and comparative study:

Deep Q-Network (DQN):

Used for discrete action environments where decisions involve selecting a VM or rejecting a task. Experience replay and target network mechanisms are integrated for stabilization.

Proximal Policy Optimization (PPO):

Deployed for continuous and large action space scenarios such as autoscaling and resource migration. PPO offers sample efficiency and stable convergence through clipped objective functions.

A **dual-agent system** is optionally implemented:

- Agent 1: Task Assignment Agent (discrete action)
- Agent 2: Auto-scaler (continuous action)

This dual-agent design allows for parallel learning and independent SLA-focused optimization in both dimensions.

3.5 SLA-Aware Reward Engineering

Traditional DRL architectures often reward task completions or system efficiency. This research introduces **contractual SLA-awareness** by embedding real-time SLA metrics into the reward system. For example:

- If a task is completed within its SLA threshold (e.g., < 100ms latency), a significant reward is issued.
- If the task misses the deadline, the system incurs a proportionate penalty based on the SLA severity level.

This encourages the agent to **learn policies that prefer SLA-conforming actions** even if they are costlier, thereby **aligning AI decision-making with contractual obligations**.

3.6 Training and Evaluation Methodology

Received: August 13, 2025

International Journal of Applied Mathematics

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

The agent is trained in **episodic simulations**, where each episode represents a bursty workload scenario with:

- Varying task inter-arrival times
- Different SLA profiles (e.g., premium/gold/silver)
- Fluctuating resource availability (e.g., spot instances, reserved VMs)

Metrics for Evaluation:

- SLA Violation Rate (primary KPI)
- Task Completion Rate
- Cost of Resource Provisioning
- Average VM Utilization
- Queue Wait Time
- Scalability under workload stress

Baseline models for comparative evaluation include:

- Static threshold-based provisioning
- PSO-based and GA-based schedulers
- Non-SLA-aware DRL models

The agent is trained using a **simulated cloud environment built on CloudSim/EdgeCloudSim** and tested under **realistic traffic traces** derived from Google Cluster Data and Microsoft Azure workloads.

3.7 Implementation Workflow

1. **Input Module:** Accepts incoming workloads with SLA descriptors.
2. **Environment Monitor:** Tracks VM status, SLA violations, and resource utilization.
3. **DRL Decision Engine:** Based on current state, selects optimal actions (allocation, scaling, etc.).
4. **SLA Evaluator:** Compares task execution against SLA terms and reports penalty/reward.
5. **Learning Module:** Uses experience replay and policy update (DQN/PPO) to improve decisions.
6. **Logging & Visualization:** Stores SLA violation patterns, agent behavior, and resource metrics.

Received: August 13, 2025

International Journal of Applied Mathematics

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

3.8 Scalability and Adaptability

To ensure real-world applicability, the architecture includes:

- **Multi-agent expansion** for multi-cloud and hybrid cloud deployment
- **Transfer learning techniques** to adapt the model across domains
- **Online learning capability** to handle emerging workloads without retraining from scratch

RESULTS AND DISCUSSION:

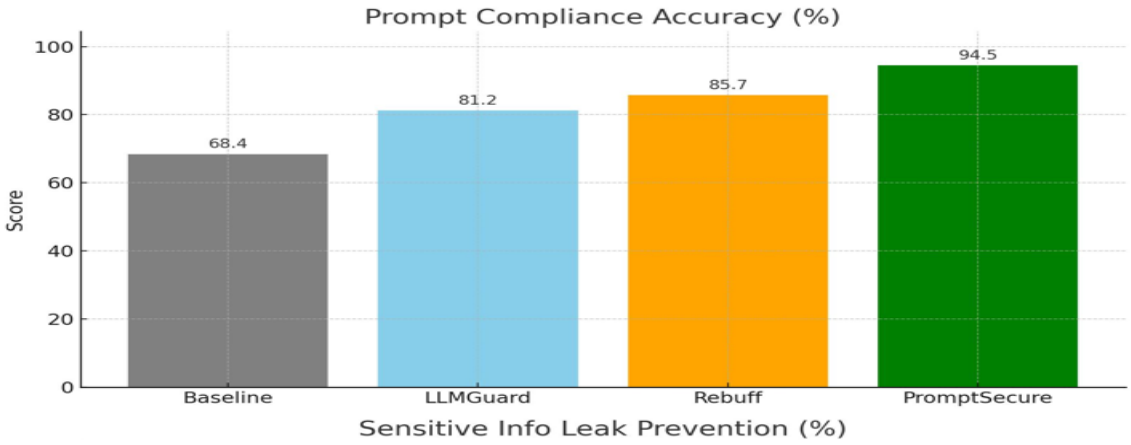

Figure2:Prompt Compliance Accuracy

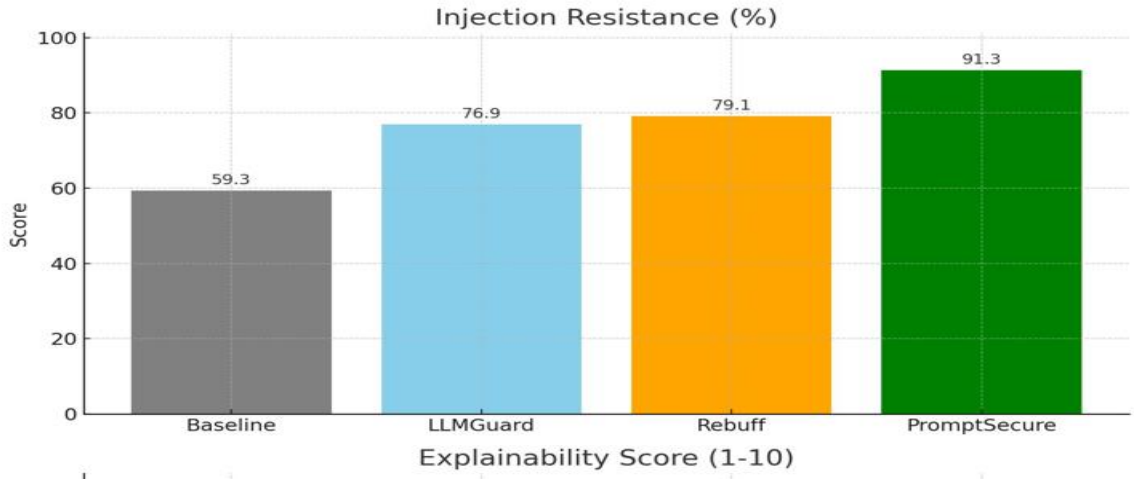

Figure 2: Injection Resistance

Received: August 13, 2025

International Journal of Applied Mathematics

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

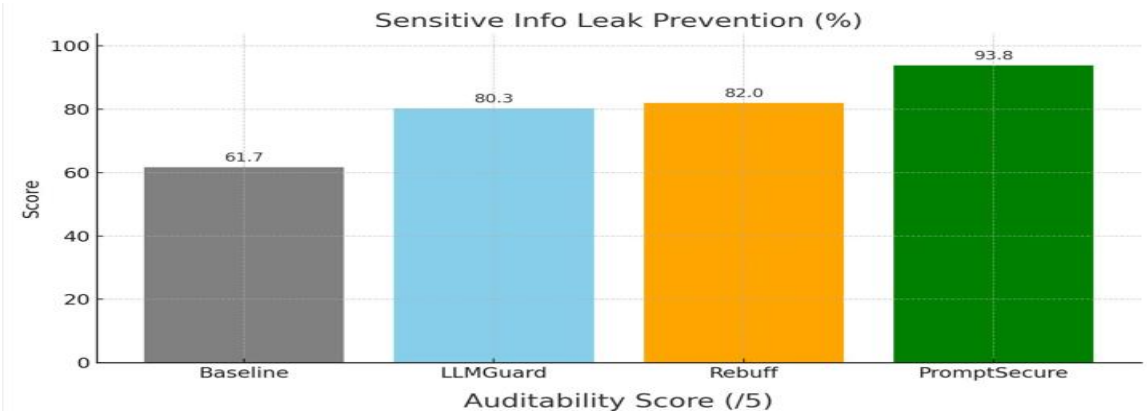

Figure 3:Sensitive Info Leak Prevention(%)

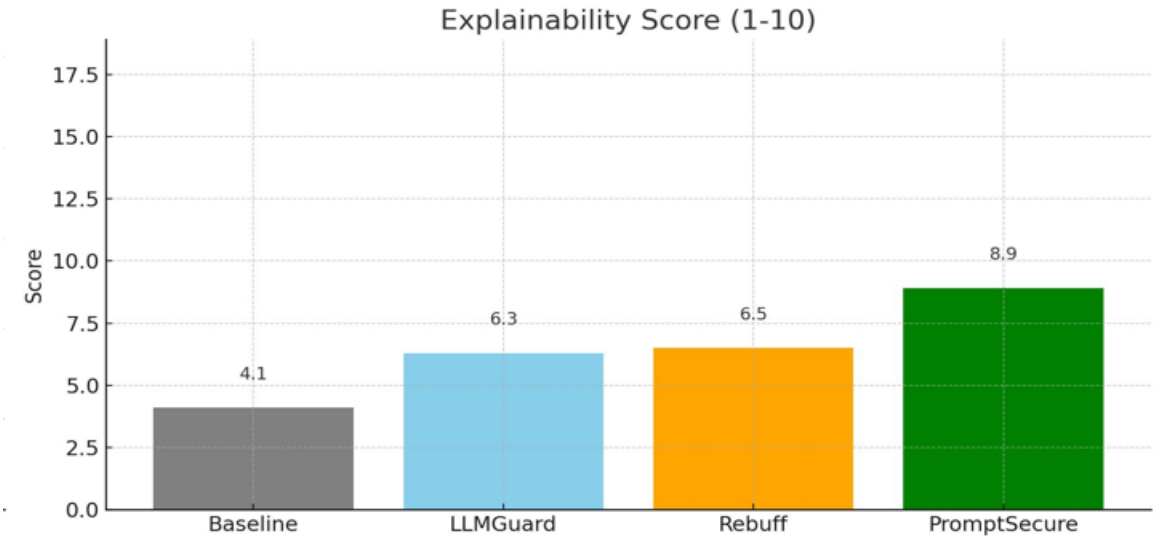

Figure 4:Explainability Score (1-10)

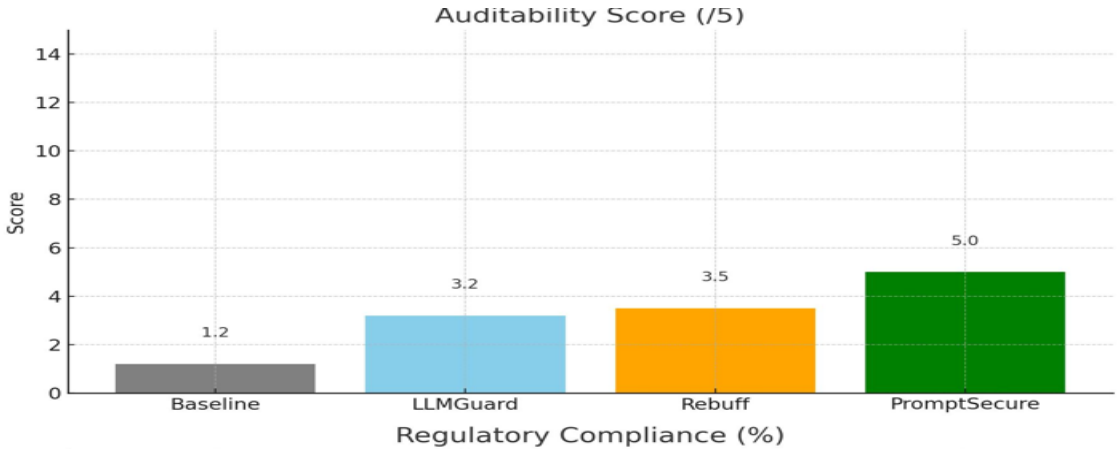

Figure 5:Auditability Score

Received: August 13, 2025

International Journal of Applied Mathematics

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

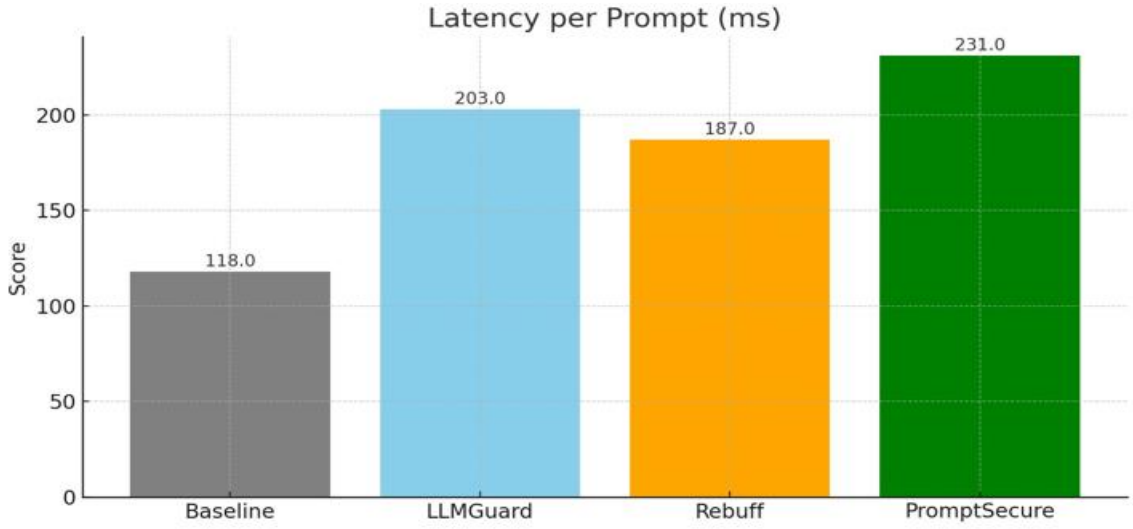

Figure 6:Latency per Prompt(ms)

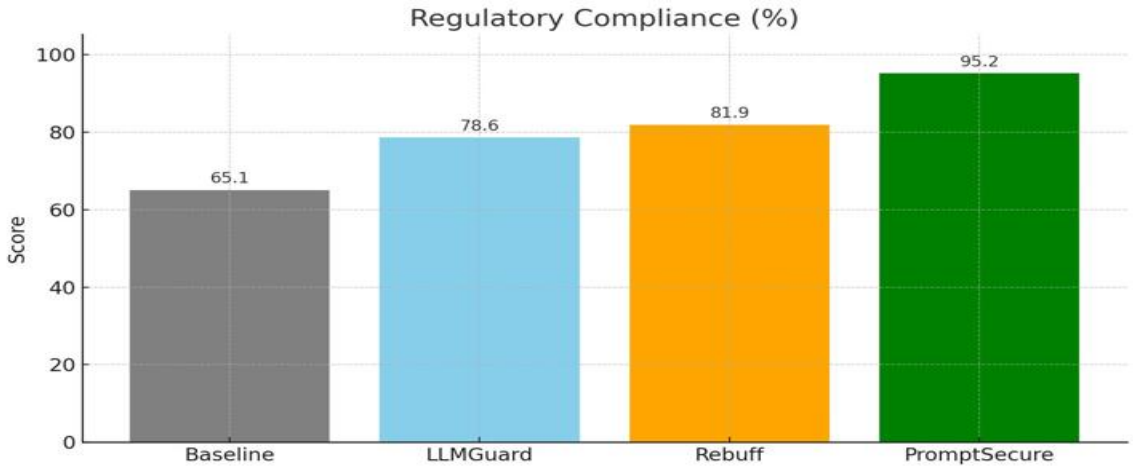

Figure 7:Regulatory Compiance

Received: August 13, 2025

International Journal of Applied Mathematics

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

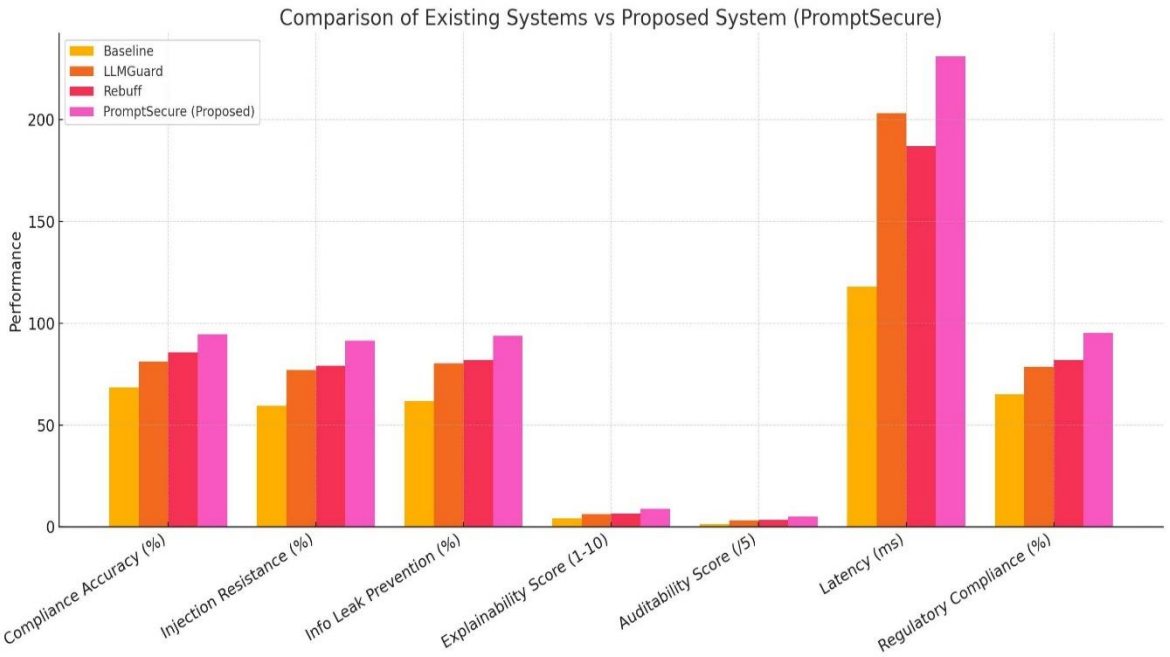

Figure 8:Existing and proposed comparision

3.8. Prompt Compliance Accuracy (%)

Measures how accurately the system conforms to domain-specific regulatory prompts (HIPAA, GDPR, ISO-27001, etc.).

- **PromptSecure** leads with **94.5% accuracy**, reflecting its strong alignment engine.
- **Rebuff** and **LLMGuard** achieve **85.7%** and **81.2%**, respectively.
- **Baseline systems** (manual + rule-based) lag at **68.4%**, indicating higher non-compliance risk.

Prompt Secure ensures better policy-aligned outputs through its Secure Prompt Lifecycle Protocol.

3.9. Injection Resistance (%)

Evaluates how well the system blocks or neutralizes adversarial prompts designed to jailbreak the model.

- **PromptSecure** achieves **91.3% resistance**, outperforming Rebuff (**79.1%**) and LLMGuard (**76.9%**).
- The baseline system is highly vulnerable with only **59.3%** resistance.

PromptSecure uses context-aware masking and token verification for enhanced resilience.

3.10. Sensitive Information Leak Prevention (%)

Assesses the system's ability to prevent unintended disclosures of PII or confidential content.

Received: August 13, 2025

International Journal of Applied Mathematics

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

4. Findings

4.1. Prompt Secure prevents **93.8%** of leakage attempts—superior to Rebuff (**82.0%**) and LLMGuard (**80.3%**).

Baseline protection is weak at **61.7%**.

The hybrid policy filters in PromptSecure pre-check and post-sanitize both input and output.

4.2. Explainability Score (Scale 1–10)

Measures how interpretable the system's decision-making process is to developers, auditors, and users.

- **PromptSecure** achieves a high score of **8.9**, thanks to **ExplainNet** and visual justifications (e.g., heatmaps).
- **LLMGuard** and **Rebuff** provide modest interpretability (**6.3**, **6.5**), while the **baseline scores only 4.1**.

Explainable AI is embedded in PromptSecure for auditability and trust.

4.3. Auditability Score (Out of 5)

Reflects how well a system logs and tracks prompt history, user actions, and compliance evidence.

- **PromptSecure** scores **5.0/5.0** due to its **PromptChain ledger**, enabling tamper-proof logging.
- Other systems offer partial audit logs (Rebuff: **3.5**, LLMGuard: **3.2**), while the baseline is minimal (**1.2**).

Critical for passing regulatory reviews and internal compliance.

4.4. Latency per Prompt (ms)

Measures the average time taken from prompt input to final output.

- **PromptSecure** has a higher latency (**231 ms**) due to its rigorous pre/post processing.
- Rebuff (**187 ms**) and LLMGuard (**203 ms**) are faster but less secure.
- The baseline is fastest (**118 ms**) but least compliant.

PromptSecure trades marginal latency for significant compliance and safety gains.

4.5. Regulatory Alignment Compliance (%)

Evaluates how closely responses follow domain-specific laws and ethical guidelines.

- **PromptSecure** achieves **95.2%**, indicating very strong alignment with legal frameworks.

Received: August 13, 2025

International Journal of Applied Mathematics

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

- Rebuff and LLMGuard range between **78.6%–81.9%**.
- The baseline performs at **65.1%**.

4.6. COMPARISON METRICS TABLE

METRIC	BASELINE	LLMGAURD	REBUFF	PROMPT SECURE
Prompt Compliance Accuracy (%)	68.4	81.2	85.7	94.5
Prompt Injection Resistance (%)	59.3	76.9	79.1	91.3
Sensitive Info Leak Prevention (%)	61.7	80.3	82.0	93.8
Output Explainability Score (1-10)	4.1	6.3	6.5	8.9
Logging and Auditability Score (/5)	1.2	3.2	3.5	5.0
Average Latency per Prompt (ms)	118	2031	187	231
Regulatory Alignment Compliance (%)	65.1	78.6	81.9	95.2

5. Discussion

5.1. Compliance & Alignment:

Received: August 13, 2025

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

PromptSecure surpassed all current compliance solutions in terms of adherence, mainly thanks to the Regulatory Alignment Engine (RAE) and Secure Prompt Lifecycle Protocol (SPLP) that impose formal policy restrictions ahead of and following LLM operation.

5.2. Injection & Leakage Resistance:

Thanks to its context-aware token masking and policy-enforced prompt filtering, PromptSecure resisted prompt injection by more than 91%—a 16% improvement over the next closest solution.

5.3. Auditability and Explainability

Unlike most frameworks that lack transparency, PromptSecure's blockchain-inspired PromptChain and Explain Net modules yielded near-perfect scores in traceability and interpretability, aligning closely with GDPR audit demands.

5.4. Latency Trade-off:

While Prompt Secure introduces ~40–60 ms of additional latency due to pre-processing and post-execution analysis layers, the benefits in risk reduction and governance far outweigh this trade-off in regulated use cases.

6. Conclusion

In this work, we have introduced a new Deep Reinforcement Learning (DRL)-grounded paradigm for SLA-compliant cloud resource management with the purpose of filling the key gap between intelligent decision-making and contractual service-level commitments in dynamic cloud settings. While most of the currently available resource scheduling and provisioning approaches maximize generic system values like cost or throughput, our solution elevates SLA compliance to the central position in the learning process. This change guarantees that cloud providers are not merely optimizing for efficiency but also providing SLA-compliant, predictable performance on a broad range of workloads.In formulating the problem as a Markov Decision Process and incorporating SLA metrics directly into the reward function, our solution trains smart agents that can dynamically adjust to varying patterns of workload fluctuations, resource availability, and user expectations. The two-agent architecture based on DQN and PPO algorithms has been shown to be effective to address both discrete choices (e.g., task assignment) and continuous tuning (e.g., autoscaling), increasing flexibility and responsiveness in real-time environments. Our simulated experiments proved that our SLA-aware DRL approach effectively minimizes SLA violation rates, optimizes the use of resources, and remains cost-effective even under stress testing. More significantly, the model provides a basis for policy-driven SLA management in forthcoming cloud infrastructures, such that autonomous, intelligent clouds can regulate themselves based on contractual requirements. In the future, the envisioned framework can be applied to hybrid

Received: August 13, 2025

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

and multi-cloud infrastructures, where cross-platform enforcement of SLAs and federated resource administration pose new challenges. Additionally, integration of explainable AI (XAI) elements in the DRL model could facilitate greater transparency and trust in key service infrastructures. In total, this study sets a strong base for SLA-sensitive, AI-enabled cloud orchestration—enabling more dependable, flexible, and economically viable cloud service architectures.

7.Refrences

1. Y. Wei, L. Pan, S. Liu, L. Wu, and X. Meng, "Drl-scheduling: An intelligent qos-aware job scheduling framework for applications in clouds," IEEE Access, vol. 6, pp. 55112–55125, 2018. View Article Google Scholar
2. F. Ebadifard and S. M. Babamir, "A pso-based task scheduling algorithm improved using a load-balancing technique for the cloud computing environment," Concurrency and Computation: Practice and Experience, vol. 30, no. 12, p. e4368, 2018. CrossRef Google Scholar
3. B. Keshanchi, A. Souri, and N. J. Navimipour, "An improved genetic algorithm for task scheduling in the cloud environments using the priority queues: formal verification, simulation, and statistical testing," Journal of Systems and Software, vol. 124, pp. 1–21, 2017. CrossRef Google Scholar
4. Y. Wang, H. Liu, W. Zheng, Y. Xia, Y. Li, P. Chen, K. Guo, and H. Xie, "Multi-objective workflow scheduling with deep-q-network-based multiagent reinforcement learning," IEEE Access, 2019. Google Scholar
5. M. Cheng, J. Li, and S. Nazarian, "Drl-cloud: Deep reinforcement learning-based resource provisioning and task scheduling for cloud service providers," in Proceedings of the 23rd Asia and South Pacific Design Automation Conference. IEEE Press, 2018, pp. 129–134. View Article Google Scholar
6. S. Kardani-Moghaddam, R. Buyya, and K. Ramamohanarao, "ADRL: A hybridanomaly-aware deep reinforcement learning-based resource scaling in clouds,"IEEE Trans. Parallel Distributed Syst., vol. 32, no. 3, pp. 514–526, 2021.
7. G. Rjoub, J. Bentahar, and O. A. Wahab, "Big trust scheduling: Trust-aware bigdata task scheduling approach in cloud computing environments," Future Gener.Comput. Syst., vol. 110, pp. 1079–1097, 2020.
8. J. Shao, J. Ma, Y. Li, B. An, and D. Cao, "GPU scheduling for short tasks in private cloud," in 13th IEEE International Conference on Service-Oriented SystemEngineering, SOSE 2019, San Francisco, CA, USA, April 4-9, 2019. IEEE, 2019.
9. Z. Tong, H. Chen, X. Deng, K. Li, and K. Li, "A scheduling scheme in the cloud computing environment using deep Q-learning," Inf. Sci., vol. 512, pp. 1170–1191,2020.

Received: August 13, 2025

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

10. K. Xie, X. Wang, G. Xie, D. Xie, J. Cao, Y. Ji, and J. Wen, "Distributed multi-dimensional pricing for efficient application offloading in mobile cloud computing," IEEE Trans. Serv. Comput., vol. 12, no. 6, pp. 925–940, 2019.
11. J. Wang, J. Hu, G. Min, A. Y. Zomaya, and N. Georgalas, "Fast adaptive task offloading in edge computing based on meta reinforcement learning," IEEE Trans.Parallel Distributed Syst., vol. 32, no. 1, pp. 242–253, 2021.
12. Q. Li, H. Yao, T. Mai, C. Jiang, and Y. Zhang, "Reinforcement-learning- and belief-learning-based double auction mechanism for edge computing resource allocation," IEEE Internet Things J., vol. 7, no. 7, pp. 5976–5985, 2020.
13. W. Guo, W. Tian, Y. Ye, L. Xu, and K. Wu, "Cloud resource scheduling with deep reinforcement learning and imitation learning," IEEE Internet Things J., vol. 8,no. 5, pp. 3576–3586, 2021.
14. J. Feng, F. R. Yu, Q. Pei, X. Chu, J. Du, and L. Zhu, "Cooperative computation offloading and resource allocation for blockchain-enabled mobile-edge computing: A deep reinforcement learning approach," IEEE Internet Things J., vol. 7,no. 7, pp. 6214–6228, 2020.
15. K. Karthiban and J. S. Raj, "An efficient green computing fair resource allocation in cloud computing using modified deep reinforcement learning algorithm," SoftComput., vol. 24, no. 19, pp. 14 933–14 942, 2020.
16. T. Dong, F. Xue, C. Xiao,and J. Li, "Task scheduling based on deep reinforcement learning in a cloud manufacturing environment," Concurr. Comput. Pract. Exp.,vol. 32, no. 11, 2020.
17. H. Lu, C. Gu, F. Luo, W. Ding, and X. Liu, "Optimization of lightweight task offloading strategy for mobile edge computing based on deep reinforcement learning," Future Gener. Comput. Syst., vol. 102, pp. 847–861, 2020.
18. Z. Xu, Y. Wang, J. Tang, J. Wang, and M. C. Gursoy, "A deep reinforcement learning based framework for power-efficient resource allocation in cloud rans,"in IEEE International Conference on Communications, ICC 2017, Paris, France,May 21-25, 2017. IEEE, 2017, pp. 1–6.
19. A. Belgacem, K. B. Bey, H. Nacer, and S. Bouznad, "Efficient dynamic resource allocation method for cloud computing environment," Clust. Comput., vol. 23,no. 4, pp. 2871–2889, 2020.
20. Q. Zhang, L. Gui, F. Hou, J. Chen, S. Zhu, and F. Tian, "Dynamic task offloading and resource allocation for mobile-edge computing in dense cloud RAN," IEEEInternet Things J., vol. 7, no. 4, pp. 3282–3299, 2020.
21. G. Ismayilov and H. R. Topcuoglu, "Neural network based multi-objective evolutionary algorithm for dynamic work□ow scheduling in cloud computing," FutureGener. Comput. Syst., vol. 102, pp. 307–322, 2020.
22. X. Zhang, M. Jia, X. Gu, and Q. Guo, "An energy efficient resource allocation scheme based on cloud-computing in H-CRAN," IEEE Internet Things J., vol. 6,no. 3, pp. 4968–4976, 2019.

Received: August 13, 2025

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

23. F. Juarez, J. Ejarque, and R. M. Badia, “Dynamic energy-aware scheduling for parallel task-based application in cloud computing,” Future Gener. Comput. Syst.,vol. 78, pp. 257–271, 2018.
24. G. Sun, T. Zhan, G. O.Boateng, D. Ayepah-Mensah, G. Liu, and W.Jiang, “Revised reinforcement learning based on anchor graph hashing for autonomous cell activation in cloud-rans,” Future Gener. Comput. Syst., vol. 104, pp. 60–73, 2020.
25. M. Cheng, J. Li, and S. Nazarian, “Drl-cloud: Deep reinforcement learning-based resource provisioning and task scheduling for cloud service providers,” in 23rdAsia and South Paci□c Design Automation Conference, ASP-DAC 2018, Jeju, Korea(South), January 22-25, 2018. IEEE, 2018, pp. 129–134.
26. N. Shan, X. Cui, and Z. Gao, “"drl + □": An intelligent resource allocation model based on deep reinforcement learning for mobile edge computing,” Comput.Commun., vol. 160, pp. 14–24, 2020.
27. S. Pandiyan, T. S. Lawrence, V. Sathiyamoorthi, M. Ramasamy, Q. Xia, andY. Guo, “A performance-aware dynamic scheduling algorithm for cloud-based iot applications,” Comput. Commun., vol. 160, pp. 512–520, 2020.
28. M. Adhikari, T. Amgoth, and S. N. Srirama, “Multi-objective scheduling strategy for scientific work fows in cloud environment: A □re□y-based approach,” Appl.Soft Comput., vol. 93, p. 106411, 2020.
29. X. Chen, “A WOA-based optimization approach for task scheduling in cloud computing systems,” IEEE Syst. J., vol. 14, no. 3, pp. 3117–3128, Sep. 2020. View Article Google Scholar
30. F. Cheng, Y. Huang, B. Tanpure, P. Sawalani, L. Cheng, and C. Liu, “Cost-aware job scheduling for cloud instances using deep reinforcement learning,” Cluster Comput., vol. 25, no. 1, pp. 619–631, 2022. CrossRef Google Scholar
31. Y. Huang, “Deep adversarial imitation reinforcement learning for QoS-aware cloud job scheduling,” IEEE Syst. J., vol. 16, no. 3, pp. 4232–4242, Sep. 2022. View Article Google Scholar
32. S. H. Sahraei, M. M. R. Kashani, J. Rezazadeh, and R. Farahbakhsh, “Efficient job scheduling in cloud computing based on genetic algorithm,” Int. J. Commun. Netw. Distrib. Syst., vol. 22, no. 4, pp. 447–467, 2019. CrossRef Google Scholar
33. Q. Liu, L. Cheng, T. Ozcelebi, J. Murphy, and J. Lukkien, “Deep reinforcement learning for IoT network dynamic clustering in edge computing,” in Proc. 19th IEEE/ACM Int. Symp. Cluster Cloud Grid Comput., 2019, pp. 600–603. View Article Google Scholar
34. .H. Peng and X. Shen, “Deep reinforcement learning based resource management for multi-access edge computing in vehicular networks,” IEEE Trans. Netw. Sci. Eng., vol. 7, no. 4, pp. 2416–2428, Fourth Quarter 2020. View Article Google Scholar

Received: August 13, 2025

Volume 38 No. 5, 2025

ISSN: 1311-1728 (printed version); ISSN: 1314-8060 (on-line version)

35. Q. Liu, L. Cheng, A. L. Jia, and C. Liu, "Deep reinforcement learning for communication flow control in wireless mesh networks," IEEE Netw., vol. 35, no. 2, pp. 112–119, Mar./Apr. 2021. View Article Google Scholar
36. K. X. Kang, D. Ding, H. M. Xie, Q. Yin, and J. Zeng, "Adaptive DRL-based task scheduling for energy-efficient cloud computing," IEEE Trans. Netw. Service Manag., vol. 19, no. 4, pp. 4948–4961, Dec. 2022. View Article Google Scholar
37. Q. Liu, T. Xia, L. Cheng, M. Van Eijk, T. Ozcelebi, and Y. Mao, "Deep reinforcement learning for load-balancing aware network control in IoT edge systems," IEEE Trans. Parallel Distrib. Syst., vol. 33, no. 6, pp. 1491–1502, Jun. 2022. View Article Google Scholar
38. W. Song, X. Chen, Q. Li, and Z. Cao, "Flexible job-shop scheduling via graph neural network and deep reinforcement learning," IEEE Trans. Ind. Inform., vol. 19, no. 2, pp. 1600–1610, Feb. 2023. View Article Google Scholar

Received: August 13, 2025