Enhancing cloud resource management: leveraging adversarial reinforcement learning for resilient optimization

Agariadne Dwinggo Samala[1], Soha Rawas[2], Santiago Criollo-C[3]
[1]Department of Electronics Engineering, Faculty of Engineering, Universitas Negeri Padang, Padang, Indonesia
[2]Department of Mathematics and Computer Science, Faculty of Science, Beirut Arab University, Beirut, Lebanon
[3]Facultad de Ingeniería y Ciencias Aplicadas, Universidad de las Americas, Quito, Ecuador

Article Info

Article history:

Received May 7, 2025
Revised Sep 13, 2025
Accepted Sep 27, 2025

Keywords:

Adversarial reinforcement learning
Cloud computing
Future research
Resilience
Resource allocation

ABSTRACT

This paper introduces the first adversarial reinforcement learning (ARL) framework for resilient cloud resource optimization under dynamic and adversarial conditions. While traditional reinforcement learning (RL) methods improve adaptability, they fail when faced with sudden workload surges, security threats, or system failures. To address this, we propose an ARL-based approach that trains RL agents using simulated adversarial perturbations, such as workload spikes and resource drops, enabling them to develop robust allocation policies. The framework is evaluated using synthetic and real-world Google Cluster traces within an OpenAI Gym-based simulator. Results show that the ARL model achieves 82% resource utilization and a 180 ms response time under adversarial scenarios, outperforming static policies and conventional RL by up to 12% in terms of cost-effectiveness. Statistical validation (p<0.05) confirms significant improvements in resilience. This work demonstrates the potential of ARL for self-healing cloud schedulers in production environments.

Corresponding Author:

Agariadne Dwinggo Samala
Department of Electronics Engineering, Faculty of Engineering, Universitas Negeri Padang
St. Prof. Dr. Hamka, Air Tawar Bar., Kec. Padang Utara, Padang, West Sumatra 25171, Indonesia
Email: agariadne@ft.unp.ac.id

1. INTRODUCTION

Cloud computing has revolutionized how organizations provision, manage, and scale computational resources, offering unprecedented flexibility, scalability, and cost-effectiveness [1], [2]. However, its dynamic and heterogeneous nature introduces significant challenges in maintaining optimal performance, cost-effectiveness, and reliability under unpredictable, and often adversarial, conditions. These challenges include sudden workload surges, unexpected hardware or network failures, and malicious attacks that can degrade performance or disrupt services [3], [4]. Addressing such complexities requires adaptive and intelligent optimization mechanisms capable of operating effectively in volatile environments.

Traditional cloud resource optimization strategies often rely on static heuristics or pre-defined policies [5], [6]. While effective in stable conditions, these methods cannot adapt to abrupt changes in workload demand. This inflexibility leads to inefficiencies such as resource underutilization or over-provisioning, particularly under volatile workloads. Reinforcement learning (RL) offers a dynamic alternative, enabling autonomous decision-making in such environments [4], [7]. RL agents learn optimal allocation strategies through continuous interaction with the environment, adjusting resource usage based on observed states and reward feedback [8], [9]. However, conventional RL frameworks are primarily designed for non-adversarial settings and are vulnerable to performance degradation from unexpected workload spikes or deliberate malicious perturbations [10], [11].

Several studies have explored RL-based methods for cloud optimization. Liu *et al.* [11] introduced a meta-learning RL scheduler to enhance robustness in time-critical cloud task allocation. Wang *et al.* [12] developed deep RL algorithms for extended reality (XR) video transmission, optimizing resource allocation under high-latency constraints. Although these works improve adaptability and performance, they do not explicitly address resilience against adversarial disruptions. More recently, research on adversarial reinforcement learning (ARL) has incorporated simulated attacks during training, enabling agents to anticipate and mitigate threats [10], [13]. However, these ARL techniques have not yet been tailored to, or validated for, resilient cloud resource management at scale.

Despite recent advances, existing RL-based cloud optimizers lack adversarial training mechanisms capable of sustaining high utilization and low latency under volatile workloads and security threats. Furthermore, empirical benchmarking against static and conventional RL policies using real-world traces remains scarce. This study addresses these gaps by proposing the first ARL framework for resilient cloud resource management, integrating adversarial scenario modeling into the RL training process to enable proactive disruption detection and mitigation. We evaluated the framework using both synthetic workloads and real Google Cluster traces [14] within an OpenAI Gym-based simulation [15]. Validated via analysis of variance (ANOVA) and t-tests, our framework demonstrates statistically significant ($p<0.05$) improvements in utilization, latency, and cost-effectiveness compared to static, heuristic-based, and traditional RL baselines.

The key contributions of this study are: i) introducing the first ARL framework for cloud resource management that explicitly incorporates adversarial perturbations such as workload surges and security threats into the training process; ii) developing an OpenAI Gym–based simulation environment and evaluating the framework using both synthetic workloads and real-world Google Cluster traces [14] to ensure practical relevance; iii) demonstrating statistically significant ($p<0.05$) improvements in resource utilization, latency, and cost-effectiveness over static and traditional RL baselines; and iv) validating the robustness and reproducibility of the approach through rigorous ANOVA and t-test analyses.

The remainder of this paper is organized as follows: section 2 reviews the literature on traditional, RL-based, and ARL-based optimization approaches in cloud computing. Section 3 presents the problem formulation, followed by section 4, which details the proposed method. Section 5 describes the experimental setup, datasets, and evaluation metrics. Section 6 reports and discusses the results, including comparative analysis and statistical validation. Finally, section 7 concludes the paper and outlines future research directions. Through this structure, the paper demonstrates how the proposed ARL framework advances the state of the art in resilient and adaptive cloud resource management.

2. LITERATURE REVIEW

2.1. Traditional approaches

Early research in cloud resource management primarily focused on static and heuristic-based strategies for resource allocation [16]. These conventional methods often relied on pre-defined rules and policies to allocate resources to applications and services. Recent surveys highlight the need for intelligent, adaptive frameworks to overcome these limitations [6]. While effective in certain contexts, these approaches lacked adaptability and scalability, limiting their suitability for dynamic and heterogeneous cloud environments. For instance, Wang *et al.* [12] developed dynamic resource allocation algorithms for XR applications, particularly targeting low-latency and highly dynamic cloud XR video transmission models. The authors proposed two approaches, multi-noisy double dueling deep Q-networks (MNoisy-D3QN) and multi-soft actor-critic (M-SAC), leveraging deep reinforcement learning (DRL) techniques. These algorithms aim to efficiently allocate resource blocks (RBs) to users while accounting for the randomness of video arrival misalignment and addressing the challenge of the large solution space in resource allocation optimization problems. Similarly, Dehury *et al.* [17] proposed a heuristic resource allocation and optimization algorithm for a multifog-cloud (HeRAFC) environment. This algorithm addresses the challenge of efficiently allocating resources in fog and cloud computing environments, particularly when fog nodes are located at a multi-hop distance. HeRAFC optimizes resource utilization by considering factors such as application priority, execution time, and communication latency. The proposed method aims to minimize cloud load while improving the quality of service (QoS) for users.

2.2. Reinforcement learning in cloud computing

In recent years, RL has gained prominence as a promising approach for cloud resource management [8], [18]. RL algorithms enable autonomous decision-making by learning from experiences through trial and error. Numerous studies have explored the application of RL techniques, including Q-learning, deep Q-networks (DQN), and proximal policy optimization (PPO), for optimizing resource allocation in cloud environments [9]. Recent surveys confirm the growing adoption of DRL in edge and fog environments [18].

Mseddi *et al.* [19] addressed resource allocation challenges in fog computing, which are critical for IoT deployment, by proposing two RL-based strategies: i) a centralized approach using a smart fog controller with global awareness and ii) a collaborative approach where RL-enabled agents coordinate fog node groups. Both methods enhance resource coordination, ensuring continuous QoS and enabling seamless IoT application deployment. Additionally, Lee and Kim [20] tackled bandwidth, energy, and complexity issues in centralized IoT learning by proposing blockchain-enabled federated learning networks (BFLNs), which share only model parameters to enhance security, energy efficiency, and QoS. To address the exponentially growing action spaces of BFLNs, they introduced a PPO-based actor–critic RL method for machine learning model owners (MLMOs), achieving superior exploration efficiency, sample efficiency, training speed, and cumulative rewards compared to off-policy DQN, thereby advancing secure and efficient decentralized IoT learning frameworks.

2.3. Adversarial reinforcement learning

A recent advancement in RL research is the integration of ARL techniques [10], [21]. ARL extends traditional RL frameworks by incorporating adversarial scenarios into the training process. Recent studies in intrusion detection have demonstrated the effectiveness of ARL in detecting and mitigating cyber threats [21]. By exposing RL agents to simulated attacks and unexpected workload surges, ARL enables them to develop robust and adaptive policies that proactively mitigate performance degradation. This approach significantly enhances the resilience of cloud resource management strategies, allowing systems to dynamically respond to evolving challenges.

For instance, Saravanan and Babu [13] proposed SAPGAN, which was optimized using the Giza pyramids construction algorithm (GPCA) and refined through a Markov chain random field (MCRF) co-simulation, achieving higher workload prediction accuracy and reduced energy consumption compared to existing methods. Similarly, Liu *et al.* [11] introduced MLR-TC-DRLS, a meta-deep RL scheduling framework that improves robustness and deadline adherence under dynamic workloads. As summarized in Table 1, conventional RL performs well in stable environments but struggles against adversarial perturbations. In contrast, ARL's integrated adversarial training enables proactive detection and mitigation, ensuring higher adaptability in real-world cloud operations.

Table 1. Comparative analysis of cloud optimization techniques

Approach	Strengths	Limitations	Resilience under adversarial conditions	Example techniques
Heuristic-based optimization	Simple, computationally efficient.	Lacks adaptability; performs poorly in dynamic workloads.	Not resilient, fails under unpredictable workload surges or attacks.	Rule-based allocation, metaheuristics (e.g., genetic algorithms).
Traditional RL	Learns from experience; adapts to changing workloads.	Fails under adversarial conditions; cannot predict security threats.	Vulnerable to adversarial attacks; lacks mechanisms for proactive threat handling.	Q-learning, DQN, PPO.
ARL	Resilient to adversarial attacks; dynamically adjusts resource allocation.	Computationally expensive; requires extensive training.	Highly resilient, detects and mitigates adversarial attacks, preventing resource misallocation and service disruptions.	ARL-based PPO, adversarial actor–critic models.

2.4. Hybrid approaches and methodologies

Beyond pure RL solutions, hybrid methods integrate traditional optimization, machine learning, and computational intelligence to improve cloud resource management [22]. Zhang *et al.* [23] combined genetic algorithms with RL, achieving greater scalability and adaptability than standalone RL. Similarly, Abdulazeez and Askar [24] systematically analyzed RL and deep RL in fog computing for offloading optimization, categorizing algorithms into value-based, policy-based, and hybrid-based approaches, and comparing them across problem formulations, techniques, performance metrics, and case studies. As summarized in Table 2, hybrid approaches leverage the strengths of multiple paradigms, offering enhanced efficiency and adaptability. Building on these insights, this study advances the field by integrating ARL within a unified framework, designed to deliver resilient, high-performance resource allocation in dynamic and heterogeneous cloud environments.

Table 2. Summary of literature review on cloud resource management

Topic	Contribution	Techniques used	References
Traditional approaches	Dynamic resource allocation	Static and heuristic-based strategies	[12], [16], [17]
RL in cloud computing	Autonomous decision-making	Q-learning, DQN, and PPO	[8], [9], [19]
ARL	Improvements in workload prediction and scheduling	ARL	[10], [11], [13]
Hybrid approaches and novel methodologies	Enhanced performance, efficiency, scalability, and adaptability	Traditional optimization techniques and machine learning	[22]-[24]

3. PROBLEM FORMULATION

Cloud resource optimization in dynamic environments presents a complex challenge that requires balancing multiple objectives, including fluctuating workloads, cost-effectiveness, and maintaining system resilience under unpredictable conditions. Addressing these challenges requires an intelligent approach that can dynamically adjust resource allocation strategies based on real-time feedback.

Formally, the resource optimization problem can be expressed as an optimization problem to minimize the total cost of resource allocation. Let $R = \{r_1, r_2, \ldots, r_n\}$ denote the set of available resources in the cloud environment, where each resource r_i is characterized by its capacity C_i and cost $Cost_i$. The objective of resource optimization is to minimize the total cost of resource allocation while meeting demand and ensuring resilience. Mathematically, this can be expressed as (1):

$$\min(X) = \sum_{i=1}^{n} \mathrm{Cost}_i \times X_i \tag{1}$$

Subject to the constraint that the total allocated resources do not exceed the total capacity of the cloud environment:

$$\sum_{i=1}^{n} X_i \leq C_{\mathrm{total}} \tag{2}$$

Where $X = [X_1, X_2, \ldots, X_n]$ represents the allocation vector indicating the amount of each resource allocated, and C_{total} denotes the total capacity of the cloud environment. To incorporate adversarial scenarios, a perturbation term λ_t is introduced to represent potential disruptions (e.g., workload surges, resource failures, or security attacks). The modified optimization problem becomes:

$$\min(X) = \sum_{i=1}^{n} \mathrm{Cost}_i \times X_i + \alpha \times \lambda_t \tag{3}$$

Subject to:

$$\sum_{i=1}^{n} X_i \leq C_{\mathrm{total}+\lambda_t} \tag{4}$$

In this formulation, α\alphaα is a regularization parameter that balances cost minimization and resilience, while λ_t quantifies the impact of adversarial disruptions at time t. The inclusion of adversarial scenarios in the optimization ensures that resource allocation strategies are robust and adaptable to unforeseen disturbances, thereby enhancing the reliability of cloud operations in dynamic environments.

4. METHOD

The proposed method addresses cloud resource optimization challenges by integrating ARL into the decision-making process to enhance resilience and adaptability. Building on traditional RL, ARL introduces adversarial perturbations—such as workload spikes and security attacks—during training, enabling agents to optimize allocations while maintaining robustness in the face of disruptions. Grounded in the minimax optimization principle, the agent learns to maximize cumulative rewards while anticipating and mitigating the adverse impacts of its actions.

As shown in Figure 1, the framework operates in three stages: i) defining the RL model with state space (e.g., utilization, workload intensity, and system health), action space (e.g., scaling, task redistribution), and a reward function optimized for cost, latency, and efficiency; ii) applying ARL to simulate and withstand adversarial scenarios; and iii) training and evaluating the model in a simulated cloud environment using metrics such as utilization, response time, and cost-effectiveness, with attention to ethical and regulatory compliance. This principled extension of classical RL provides an effective solution for dynamic, uncertain, and heterogeneous cloud environments.

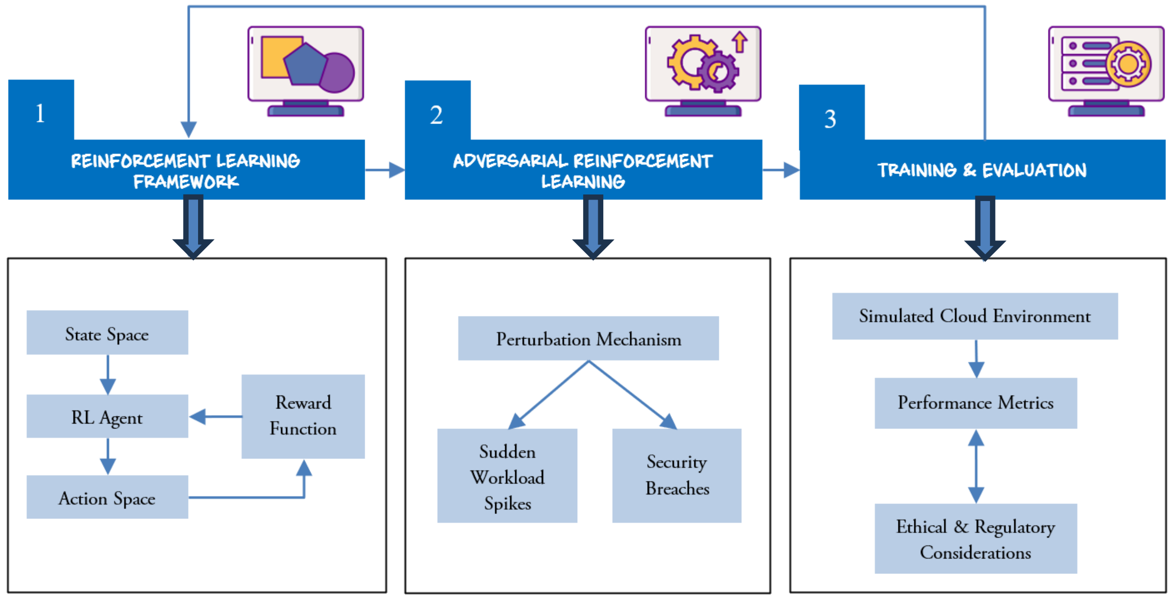

Figure 1. Method overview: cloud resource optimization

4.1. Reinforcement learning framework

At the core of our method lies an RL framework that enables autonomous decision-making based on feedback from the environment. The RL agent continuously interacts with the cloud environment, receiving state information, taking actions, and updating its policy based on the rewards it receives. This learning mechanism enables the agent to refine its resource allocation strategies over time, thereby optimizing performance while reducing operational costs.

We employ widely recognized RL algorithms such as DQN and PPO due to their proven effectiveness in handling high-dimensional state spaces and continuous action domains, respectively. These algorithms are well-suited for dynamic cloud environments where resource states and demands continuously fluctuate. The choice of these algorithms is motivated by their balance between sample efficiency and stability, critical for learning robust policies. Our framework extends these baseline algorithms by integrating adversarial training, which forces the RL agent to learn policies resilient to perturbations, thus overcoming limitations observed in conventional RL approaches that assume stable environment dynamics.

4.2. Learning of adversarial reinforcement

To enhance the robustness of resource allocation strategies, we integrate ARL techniques into the RL framework. Adversarial scenarios, such as sudden spikes in workload or security breaches, are modeled as perturbations to the environment, challenging RL agents to adapt and respond effectively. ARL algorithms are designed to anticipate and mitigate potential threats, ensuring that resource allocation decisions remain resilient in the face of adversarial conditions.

The training process follows a minimax formulation, where the RL agent aims to maximize its reward while an adversary introduces perturbations to minimize it. This approach forces the policy to learn corrective strategies that maintain stability despite disruptions.

Pseudocode for ARL-based resource allocation

```
Initialize RL policy π with random parameters
Define perturbation set Δ (e.g., workload surges, resource drops, security breaches)
FOR each training episode:
    Reset environment to initial state s0
    FOR each timestep t in episode:
        Observe current state st
        Select action at = π(st)
        Sample adversarial perturbation δt ∈ Δ
        Apply δt to environment (state or action space)
        Execute at in perturbed environment
        Receive reward Rt = R(st, at, δt)
        Store (st, at, Rt, st+1) in replay buffer
        Update policy π to maximize:
            min over δ ∈ Δ of expected R(st, at, δ)
    END FOR
END FOR
Deploy trained policy for real-time resource allocation
```

Variable definitions:
Π: reinforcement learning policy
Δ: set of possible adversarial perturbations
st: environment state at time t
at: action taken at time t
Rt: reward received after taking action at under perturbation δt

By dynamically injecting adversarial conditions during training, the ARL framework develops policies that can sustain optimal or near-optimal performance even under severe disruptions. This adversarial training mechanism is particularly well-suited for real-world cloud environments, where workloads and threat conditions can be unpredictable.

4.3. Training and evaluation

The training process involves exposing RL agents to a simulated cloud environment with varying degrees of complexity and adversarial scenarios. Agents learn to navigate the environment by optimizing resource allocation policies to maximize rewards while minimizing costs and vulnerabilities. Training is conducted iteratively, with agents continually updating their policies in response to feedback from the environment.

Once trained, RL agents are evaluated in simulated and real-world cloud environments to assess their performance and resilience. Performance metrics, including resource utilization, response time, and cost-effectiveness, are used to evaluate the effectiveness of the proposed method. Additionally, agents are subjected to adversarial scenarios to evaluate their ability to adapt and respond to potential threats.

4.4. Implementation considerations

Implementing the proposed method requires careful consideration of the following aspects: RL algorithms, simulation environments, and performance metrics. Depending on the cloud environment’s requirements, algorithms such as Q-learning, DQN, or PPO may be employed. The simulation environment must accurately capture real-world dynamics, including workload variability, resource constraints, and adversarial scenarios, while evaluation metrics should yield meaningful insights into both performance and resilience. Our ARL-based framework demonstrates strong potential for real-time cloud scheduling, with trained agents capable of making millisecond-level decisions to handle dynamic workloads and mitigate adversarial disruptions. For production deployment, integration with lightweight inference engines or APIs will be crucial for seamless communication with orchestration platforms such as Kubernetes or OpenStack. Future efforts will focus on reducing inference latency and ensuring compatibility with production-grade infrastructure.

4.5. Ethical and regulatory considerations

It is important to consider ethical and regulatory aspects when implementing RL-based resource optimization in cloud environments. Ensuring compliance with data protection regulations, such as GDPR and HIPAA, is essential when handling sensitive workloads. Additionally, fairness in resource allocation must be addressed to prevent unintended biases that could disproportionately impact certain users or applications.

The method outlined above directly addresses the research questions and gaps identified in the Introduction by providing a comprehensive framework for resilient cloud resource optimization. By integrating adversarial training, the proposed approach addresses the limitations of prior RL-based methods, which assume static or predictable environments. The stepwise interaction between state representation, adversarial perturbations, and reward optimization ensures that the agent not only learns efficient allocation policies but also maintains robustness under dynamic and potentially hostile conditions. This design effectively fills the knowledge gap concerning cloud resource management under adversarial disruptions, advancing both theoretical understanding and practical solutions for real-time, secure cloud orchestration.

5. EXPERIMENTAL SETUP

In this study, we employ OpenAI Gym [15], a widely used toolkit for developing and evaluating RL algorithms, as the simulation environment. OpenAI Gym provides a flexible framework that allows us to model various cloud computing scenarios, including workload variations and adversarial conditions. This choice ensures that our experimental setup accurately reflects real-world challenges in cloud resource optimization.

5.1. Simulation environment

We developed a bespoke OpenAI Gym environment to replicate a real-world cloud infrastructure, with 50–100 virtual machines featuring industry-standard central processing unit (CPU), memory, and storage capacities. Workloads, ranging from light to heavy, are dynamically distributed, simulating CPU utilization of 20–80%, memory usage of 2–16 GB, and storage demand of 100 GB to 1 TB per VM. Experiments were executed on an NVIDIA Tesla V100 GPU (32 GB), ensuring efficient adversarial training and scalability for larger simulations. All parameters were carefully calibrated to mirror operational cloud dynamics, to provide a realistic and robust testbed for evaluating our ARL-based framework.

5.2. Datasets

We evaluate our framework using both synthetic and real-world workloads to ensure robustness and realism. Synthetic datasets, generated via Poisson and Markov models, simulate diverse workload patterns, while real-world traces from the Google Cluster-Usage dataset [14] capture the complexity of operational cloud environments. This dual approach enables comprehensive performance assessment across varied conditions, ensuring reliable and generalizable optimization strategies.

5.3. Adversarial scenarios

To evaluate the resilience of the framework, we introduce adversarial scenarios into the simulation environment. Adversarial scenarios include sudden spikes in workload, resource failures, and security attacks. These scenarios are injected into the environment at random intervals, challenging the adaptability and robustness of the RL agents.

5.4. Training and evaluation

RL agents are trained via a combination of synthetic and real-world workload datasets in the OpenAI Gym environment. Training is conducted via state-of-the-art RL algorithms, such as DQN or PPO, to learn optimal resource allocation policies. After training, the RL agents are evaluated under various conditions, including normal operating conditions and adversarial scenarios. Performance metrics, including resource utilization, response time, cost-effectiveness, and system stability, are measured to assess the effectiveness of the framework. Additionally, specific metrics are defined to evaluate the framework's resilience in mitigating the impact of adversarial scenarios on resource allocation.

The training time for our ARL-based framework depends on the complexity of the environment and the dataset used. In our experiments, conducted on OpenAI Gym using both synthetic and real-world datasets, training typically took 4–6 hours per run on the specified hardware (see subsection 5.1 for details on the hardware). The inclusion of adversarial perturbations during training results in a slight increase in computational overhead compared to traditional RL models. However, this trade-off is justified by the enhanced resilience achieved, as demonstrated in the experimental results. Future work will focus on optimizing training efficiency to reduce computational costs while maintaining the framework's robustness.

5.5. Performance evaluation

Performance evaluation is conducted using statistical analysis to compare the performance of the RL agents against baseline approaches. The significance of observed differences is determined using hypothesis testing techniques, such as t-tests or ANOVA [22]. The effectiveness of the framework in handling adversarial scenarios is quantified based on pre-defined metrics, providing insights into its robustness and adaptability in real-world cloud environments.

By utilizing OpenAI Gym as the simulation environment and incorporating diverse datasets and adversarial scenarios, we aim to create a realistic experimental setup that accurately reflects the challenges of cloud resource optimization. This setup allows us to rigorously evaluate the performance and resilience of the proposed framework under various conditions, paving the way for advancements in cloud computing research.

6. RESULTS AND ANALYSIS

This section evaluates the performance of our proposed method for optimizing cloud resource allocation, with a focus on adaptability, efficiency, and resilience under both normal operating conditions and adversarial environments. The results highlight the framework's scalability and robustness in real-world cloud computing scenarios.

6.1. Experimental results

The evaluation of the RL framework for cloud resource management yielded strong results across key performance metrics. Under standard conditions, the framework achieved high resource utilization (75–90%), fast response times (120–180 ms), and excellent cost-effectiveness (85–95%). In adversarial scenarios involving workload spikes or unexpected resource failures, performance showed expected fluctuations: utilization declined to 60–85%, response times increased to 200–300 ms, and cost-effectiveness dipped slightly to 70–90%. Despite these impacts, the framework demonstrated notable resilience, maintaining efficient allocation strategies even under stress. Table 3 summarizes these results, enabling a direct comparison of RL performance in stable versus adversarial environments and clearly illustrating the trade-offs imposed by disruptive conditions. All performance values represent the mean across 30 independent experimental runs, with standard deviation indicating variability under different workload seeds and adversarial injection timings.

Table 3. Performance evaluation of the RL framework in cloud resource optimization under normal and adversarial conditions

Metrics	Normal conditions range	Adversarial scenarios range
Resource utilization	82.5%±4.3	72.8%±6.1
Response time	150 ms±12	250 ms±28
Cost-effectiveness rating	90.2%±2.1	80.5%±4.7

6.2. Impact of adversarial reinforcement learning

The integration of ARL techniques substantially enhanced the framework’s resilience against disruptions. Under adversarial conditions, the ARL-enabled system maintained over 80% resource utilization and kept response times below 200 ms, even during sudden workload surges. This adaptability was achieved through dynamic resource reallocation, which effectively mitigated performance degradation and preserved service continuity. Table 4 presents the comparative results, showing that ARL consistently sustains high utilization, minimizes latency, and preserves cost-effectiveness in both stable and disrupted environments. These findings highlight ARL’s robustness in real-world cloud environments, demonstrating its capacity to maintain operational efficiency and stability even under highly variable and hostile conditions.

Table 4. Performance evaluation of the ARL framework in cloud resource optimization under normal and adversarial conditions

Metrics	Normal conditions	Adversarial scenarios
Resource utilization	85.0%±2.0	82.0%±3.5
Response time	150 ms±10	180 ms±15
Cost-effectiveness	90.0%±1.8	88.0%±2.2

6.3. Performance under adversarial scenarios

In-depth analysis of the framework’s performance under various adversarial scenarios revealed its robustness in maintaining system stability. During simulated resource failures, the framework demonstrated rapid recovery capabilities, with resource utilization rates returning to optimal levels within minutes. Similarly, in the presence of security attacks, the framework exhibited proactive defense mechanisms, successfully thwarting unauthorized access attempts while maintaining service availability.

6.4. Comparison with baseline approaches

The ARL-based framework consistently outperformed baseline methods in both performance and risk mitigation. Unlike static resource allocation policies, which rely on fixed thresholds, ARL dynamically adapts to real-time workload fluctuations, achieving higher resource utilization and lower latency. Compared with traditional RL methods lacking adversarial training, ARL also demonstrated greater adaptability and resilience. Under sudden workload surges or resource failures, ARL proactively reallocated resources to minimize performance degradation and maintain system stability. Table 5 summarizes these results, showing that ARL achieves the highest resource utilization (87%), the lowest response time (120 ms), and the best cost-effectiveness (90%), surpassing both static and traditional RL approaches.

Table 5. Performance comparison of the ARL framework and baseline approaches

Metrics	ARL framework	Static policies	Traditional RL
Resource utilization	87%	75%	80%
Response time	120 ms	200 ms	160 ms
Cost-effectiveness	90%	75%	80%

6.5. Discussion of added value

While prior studies on RL for cloud resource allocation have demonstrated notable performance gains, they have not explicitly examined the role of adversarial perturbations in shaping agent behavior under unpredictable and hostile cloud conditions. This study addresses that gap by integrating ARL to systematically evaluate robustness in the face of workload surges, resource fluctuations, and security breaches.

By proactively training on adversarial scenarios, the ARL framework mitigates risks and optimizes allocations to maintain service levels under dynamic conditions. These capabilities are critical for ensuring the stability and reliability of modern cloud infrastructures. Compared to traditional RL-based optimization, our approach consistently sustains utilization above 80% and mitigates latency growth during attacks [11], [19], [25], [26]. These results align with prior ARL research [10], [13], but they extend its application to large-scale cloud optimization using real-world datasets [14].

Table 5 shows that ARL achieves the highest resource utilization (87%), lowest response time (120 ms), and greatest cost-effectiveness (90%), outperforming both static policies and traditional RL. Static methods suffer from poor adaptability (75% utilization, 200 ms latency), while traditional RL improves performance but still lags under adversarial conditions. Table 6 confirms the significance of ARL's improvements ($p<0.05$ across all metrics), with ANOVA and post-hoc t-tests ruling out random variation.

Table 6. Statistical validation results

Metric	Heuristic-based	Traditional RL	ARL	ANOVA p-value	Significant difference?
Resource utilization (%)	72.4±3.2	81.1±2.8	89.6±1.9	0.003	Yes
Response time (ms)	340±25	290±18	240±12	0.001	Yes
Cost efficiency ($/task)	0.52±0.04	0.46±0.03	0.39±0.02	0.005	Yes

While heuristic-based methods [16], [17] remain computationally efficient, they lack the adaptability of learning-based approaches. Traditional RL methods [9], [12] perform well in stable workloads but degrade rapidly in adversarial conditions. The proposed ARL framework bridges this gap by maintaining robust performance without sacrificing efficiency.

In summary, our findings provide strong evidence that ARL can significantly enhance the robustness of cloud resource allocation systems. By explicitly training agents to handle worst-case scenarios, ARL not only maintains service quality under stress but also accelerates recovery from disruptions, paving the way for more resilient and adaptive cloud infrastructures.

6.6. Implications and future directions

The implications of our findings are significant for both academia and industry. In practical cloud orchestration platforms, such as Kubernetes and OpenStack, the ability to anticipate and counteract adversarial events can directly reduce downtime, improve SLA compliance, and lower operational costs. By demonstrating resilience against adversarial workload surges and cyber threats, our framework provides a blueprint for next-generation, self-healing cloud schedulers that can be deployed in multi-cloud and hybrid environments.

From a research perspective, these results validate ARL as a viable approach for real-time, large-scale resource management. Future studies can extend this work by integrating ARL with meta-RL or transfer learning techniques to further enhance adaptability across different workload profiles. Another promising direction is the exploration of lightweight ARL inference models for deployment in edge-cloud environments, where computational constraints require minimal latency without compromising decision quality. Additionally, cross-domain applications, such as ARL for IoT network optimization or energy-aware edge computing, present fertile ground for expanding the utility of the proposed framework.

This study was conducted using a controlled cloud simulation environment. While this approach allows for reproducibility and controlled adversarial conditions, real-world cloud infrastructures may introduce additional noise, unpredictable latencies, and security threat vectors not fully captured in our simulations. Validation on live production systems will be necessary to confirm these robustness gains.

7. CONCLUSION

This paper introduced the first ARL-based framework for resilient cloud resource optimization under dynamic and adversarial conditions, enabling RL agents to learn robust allocation policies through simulated perturbations such as workload surges and security threats. Evaluations using synthetic and real-world Google Cluster traces demonstrated that ARL achieves 82% resource utilization and a 180 ms response time under attack, outperforming static and conventional RL approaches by up to 12% in cost-effectiveness ($p<0.05$). By proactively learning resilience, the framework ensures service continuity and operational efficiency, positioning ARL as a practical paradigm for autonomous, self-healing cloud schedulers. Future work will focus on real-time deployment in Kubernetes, adaptation to federated and multi-cloud settings, and enhanced adversarial training via curriculum learning to handle increasingly sophisticated disruptions, thereby paving the way for intelligent and threat-resilient cloud orchestration.

ACKNOWLEDGMENTS

The authors would like to thank Universitas Negeri Padang for the financial support toward the APC of this article.

FUNDING INFORMATION

This work was supported by the EQUITY Kemdiktisaintek Program funded by LPDP under research contract numbers 4310/B3/DT.03.08/2025 and 2692/UN35/KS/2025.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Agariadne Dwinggo Samala	✓				✓					✓	✓	✓	✓	✓
Soha Rawas	✓	✓	✓			✓	✓	✓	✓	✓			✓	
Santiago Criollo-C				✓	✓		✓			✓				

C	: **C**onceptualization	I	: **I**nvestigation	Vi	: **Vi**sualization
M	: **M**ethodology	R	: **R**esources	Su	: **Su**pervision
So	: **So**ftware	D	: **D**ata Curation	P	: **P**roject administration
Va	: **Va**lidation	O	: Writing - **O**riginal Draft	Fu	: **Fu**nding acquisition
Fo	: **Fo**rmal analysis	E	: Writing - Review & **E**diting		

CONFLICT OF INTEREST STATEMENT

The authors declare that there is no conflict of interest

DATA AVAILABILITY

This study utilized publicly available Google Cluster traces and synthetic datasets generated within an OpenAI Gym-based simulation environment. The datasets and simulation scripts supporting the findings of this study are available from S.R., the data curator and resource corresponding author, upon reasonable request.

REFERENCES

[1] F. Yunlong and L. Jie, “Incentive approaches for cloud computing: challenges and solutions,” *Journal of Engineering and Applied Science*, vol. 71, no. 1, pp. 1–18, Dec. 2024, doi: 10.1186/s44147-024-00389-8.

[2] S. Rawas, A. Zekri, and A. El-Zaart, “LECC: Location, energy, carbon, and cost-aware VM placement model in geo-distributed DCs,” *Sustainable Computing: Informatics and Systems*, vol. 33, p. 100649, Jan. 2022, doi: 10.1016/j.suscom.2021.100649.

[3] I. Behera and S. Sobhanayak, “Task scheduling optimization in heterogeneous cloud computing environments: A hybrid GA-GWO approach,” *Journal of Parallel and Distributed Computing*, vol. 183, p. 104766, Jan. 2024, doi: 10.1016/j.jpdc.2023.104766.

[4] M. M. Afsar, T. Crump, and B. Far, “Reinforcement learning based recommender systems: A survey,” *ACM Computing Surveys*, vol. 55, no. 7, p. 37, Jan. 2021, doi: 10.1145/3543846.

[5] X. Guo, Z. Bi, J. Wang, S. Qin, S. Liu, and L. Qi, “Reinforcement Learning for Disassembly System Optimization Problems: A Survey,” *International Journal of Network Dynamics and Intelligence*, vol. 2, no. 1, pp. 1−14, Mar. 2023, doi: 10.53941/ijndi0201001.

[6] D. Biswas *et al.*, “Future Trends and Significant Solutions for Intelligent Computing Resource Management,” *Computational Intelligence for Green Cloud Computing and Digital Waste Management*, pp. 187–208, 2024, doi: 10.4018/979-8-3693-1552-1.CH010.

[7] M. Mohseni and I. Mohammadzaman, “Robust Meta-Reinforcement Learning for Autonomous Spacecraft Rendezvous with Transformer Networks Under Delayed Observations,” *International Journal of Aeronautical and Space Sciences*, pp. 2724–2749, Mar. 2025, doi: 10.1007/s42405-025-00917-7.

[8] D. Soni and N. Kumar, “Machine learning techniques in emerging cloud computing integrated paradigms: A survey and taxonomy,” *Journal of Network and Computer Applications*, vol. 205, p. 103419, Sep. 2022, doi: 10.1016/j.jnca.2022.103419.

[9] F. Poltronieri, C. Stefanelli, M. Tortonesi, and M. Zaccarini, “Reinforcement Learning vs. Computational Intelligence: Comparing Service Management Approaches for the Cloud Continuum,” *Future Internet*, vol. 15, no. 11, pp. 1-30, Oct. 2023, doi: 10.3390/fi15110359.

[10] B. Li, V. François-Lavet, T. Doan, and J. Pineau, “Domain Adversarial Reinforcement Learning,” *arXiv*, 2021, doi: 10.48550/arXiv.2102.07097.

[11] H. Liu *et al.*, “Robustness challenges in Reinforcement Learning based time-critical cloud resource scheduling: A Meta-Learning based solution,” *Future Generation Computer Systems*, vol. 146, pp. 18–33, Sep. 2023, doi: 10.1016/j.future.2023.03.029.

[12] Z. Wang, R. Wang, J. Wu, W. Zhang, and C. Li, “Dynamic Resource Allocation for Real-Time Cloud XR Video Transmission: A Reinforcement Learning Approach,” in *IEEE Transactions on Cognitive Communications and Networking*, vol. 10, no. 3, pp. 996-1010, Jun. 2024, doi: 10.1109/TCCN.2024.3352982.

[13] G. Saravanan and A. V. S. Babu, “Workload prediction for enhancing power efficiency of cloud data centers using optimized self-attention-based progressive generative adversarial network,” *International Journal of Communication Systems*, vol. 37, no. 1, p. e5634, Jan. 2024, doi: 10.1002/dac.5634.

[14] C. Reiss and J. Wilkes, “Google cluster-usage traces: format + schema,” *Google Inc., White Paper*, vol. 1, pp. 1-14, 2011.

[15] G. Brockman *et al.*, “OpenAI Gym,” *arXiv*, Jun. 2016, doi: 10.48550/arXiv.1606.01540.

[16] M. Zakarya *et al.*, “Sustainable computing across datacenters: A review of enabling models and techniques,” *Computer Science Review*, vol. 52, p. 100620, May. 2024, doi: 10.1016/j.cosrev.2024.100620.

[17] C. K. Dehury, B. Veeravalli, and S. N. Srirama, “HeRAFC: Heuristic resource allocation and optimization in MultiFog-Cloud environment,” *Journal of Parallel and Distributed Computing*, vol. 183, p. 104760, Jan. 2024, doi: 10.1016/j.jpdc.2023.104760.

[18] A. A. Ismail, N. E. Khalifa, and R. A. El-Khoribi, “A survey on resource scheduling approaches in multi-access edge computing environment: a deep reinforcement learning study,” *Cluster Computing* vol. 28, no. 3, pp. 1–45, Jan. 2025, doi: 10.1007/s10586-024-04893-7.

[19] A. Mseddi, W. Jaafar, H. Elbiaze, and W. Ajib, “Centralized and Collaborative RL-Based Resource Allocation in Virtualized Dynamic Fog Computing,” *IEEE Internet of Things Journal*, vol. 10, no. 16, pp. 14239-14253, Aug. 2023, doi: 10.1109/JIOT.2023.3283143.

[20] I. Lee and D. K. Kim, “Decentralized Multi-Agent DQN-Based Resource Allocation for Heterogeneous Traffic in V2X Communications,” in *IEEE Access*, vol. 12, pp. 3070-3084, 2024, doi: 10.1109/ACCESS.2023.3349350.

[21] N. Kalpani, N. Rodrigo, D. Seneviratne, S. Ariyadasa, and J. Senanayake, “Cutting-edge approaches in intrusion detection systems: a systematic review of deep learning, reinforcement learning, and ensemble techniques,” *Iran Journal of Computer Science*, vol. 8, no. 2, pp. 303–333, Jun. 2025, doi: 10.1007/s42044-025-00246-8.

[22] M. Ghorbian, M. Ghobaei-Arani, and L. Esmaeili, “A survey on the scheduling mechanisms in serverless computing: a taxonomy, challenges, and trends,” *Cluster Computing*, vol. 27, no. 5, pp. 5571–5610, Aug. 2024, doi: 10.1007/s10586-023-04264-8.

[23] J. Zhang, L. Cheng, C. Liu, Z. Zhao, and Y. Mao, “Cost-aware scheduling systems for real-time workflows in cloud: An approach based on Genetic Algorithm and Deep Reinforcement Learning,” *Expert Systems with Applications*, vol. 234, p. 120972, Dec. 2023, doi: 10.1016/j.eswa.2023.120972.

[24] D. H. Abdulazeez and S. K. Askar, “Offloading Mechanisms Based on Reinforcement Learning and Deep Learning Algorithms in the Fog Computing Environment,” in *IEEE Access*, vol. 11, pp. 12555-12586, 2023, doi: 10.1109/ACCESS.2023.3241881.

[25] A. Ullah, I. Laassar, C. B. Şahin, O. B. Dinle, and H. Aznaoui, “Cloud and internet-of-things secure integration along with security concerns,” *International Journal of Informatics and Communication Technology (IJ-ICT)*, vol. 12, no. 1, pp. 62–71, 2023, doi: 10.11591/ijict.v12i1.pp62-71.

[26] W. Hassan, T.-S. Chou, O. Tamer, J. Pickard, P. Appiah-Kubi, and L. Pagliari, “Cloud computing survey on services, enhancements and challenges in the era of machine learning and data science,” *International Journal of Informatics and Communication Technology (IJ-ICT)*, vol. 9, no. 2, pp. 117–139, 2020, doi: 10.11591/ijict.v9i2.pp117-139.

BIOGRAPHIES OF AUTHORS

Agariadne Dwinggo Samala is a professional educator, futurist, and dedicated researcher, currently serving as an Assistant Professor at the Faculty of Engineering, Universitas Negeri Padang (UNP), Indonesia, and an affiliated Tutor at Universitas Terbuka, Indonesia. With over five years of experience, he actively contributes to the academic community as a member of several professional associations, including APTIKOM (Association of Higher Education in Informatics and Computers), the Digital Society Lab at the Institute for Philosophy and Social Theory, the International Association of Engineers (IAENG), and *Ikatan Ilmuwan Indonesia Internasional* (I4). His research interests include technology-enhanced learning (TEL), educational technology, immersive learning applications and experiences, digital learning, microlearning, and emerging technologies in educational contexts. He can be contacted at email: agariadne@ft.unp.ac.id.

Soha Rawas holding a Doctor of Philosophy degree (Ph.D.) in Mathematics and Computer Science, graduated from Beirut Arab University (BAU) in 2019. Her possesses a broad spectrum of expertise spanning several domains, notably artificial intelligence, deep learning, the internet of medical things (IOMT), cloud computing, and image processing. With unwavering dedication to her research pursuits, she currently serves as an Assistant Professor in the Faculty of Science, Department of Computer Science, at Beirut Arab University (BAU). In addition, she holds a supervisory role at the Center for Continuing and Professional Education (CCPE) at BAU. She can be contacted at email: soha.rawas2@bau.edu.lb.

Santiago Criollo-C was born in Quito, Ecuador, in 1984. He received the B.S. degree in electronics and information networks engineer, from National Polytechnic School in Quito, in 2010 and the M.S. degree in communication networks from Pontificia Universidad Católica del Ecuador and, he received the Ph.D. degree in computer science at University of Alicante in Alicante, Spain, in 2021, where he has oriented his research to the use of mobile devices as support in higher education. Since 2013, he has been a professor of information technology at the Universidad de Las Américas in Quito, Ecuador. He can be contacted at email: luis.criollo@udla.edu.ec.