Journal of Data Analysis and Information Processing, 2025, 13(3), 255-268
https://www.scirp.org/journal/jdaip
ISSN Online: 2327-7203
ISSN Print: 2327-7211

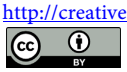
Reinforcement Learning for Dynamic and Predictive CPU Resource Management in Cloud Computing

Yihan Wang[1], Suchuan Xing[2*]

[1]School of Engineering and Applied Science, University of Pennsylvania, Philadelphia, PA, USA
[2]Department of Electrical and Computer Engineering, Duke University, Durham, NC, USA
Email: *sx80@alumni.duke.edu

How to cite this paper: Wang, Y.H. and Xing, S.C. (2025) Reinforcement Learning for Dynamic and Predictive CPU Resource Management in Cloud Computing. *Journal of Data Analysis and Information Processing*, **13**, 255-268.
https://doi.org/10.4236/jdaip.2025.133015

Received: June 18, 2025
Accepted: July 25, 2025
Published: July 28, 2025



Abstract

As cloud computing continues to evolve, managing CPU resources effectively has become a critical task for ensuring system performance and efficiency. Traditional CPU resource management methods, such as static allocation and manual optimization, are increasingly inadequate in handling dynamic, fluctuating workloads characteristic of modern cloud environments. This paper explores the use of Reinforcement Learning (RL) for adaptive CPU resource management, offering a dynamic, data-driven approach to optimizing resource allocation in real-time. Reinforcement learning, particularly Q-learning and Deep Q Networks (DQNs), enables cloud systems to autonomously adjust CPU resources based on workload demands, improving system efficiency and minimizing resource wastage. This paper discusses the key principles of reinforcement learning, its applications in CPU resource management, the benefits of its implementation, and the challenges that need to be addressed for broader adoption. Finally, the paper highlights future directions for integrating RL with other machine learning techniques and its potential impact on cloud infrastructure optimization.

Keywords

Reinforcement Learning, CPU Resource Management, Adaptive Systems, Cloud Computing, Resource Allocation, Machine Learning, Dynamic Resource Management, AI, Automation, Virtualization

1. Introduction

In today's rapidly evolving cloud computing environments, efficient CPU resource management is critical for ensuring that cloud services perform optimally [1]. As

cloud computing platforms handle increasingly diverse and dynamic workloads, traditional resource management techniques, which are often static and rely on predetermined configurations, are becoming increasingly inadequate [2]. CPU resource allocation in modern cloud systems needs to be more adaptive, capable of handling changing demands and making real-time adjustments to maintain performance and avoid resource wastage [3]. This is particularly important in multi-tenant environments where resources are shared among multiple users, and dynamic workloads are the norm.

Reinforcement Learning (RL), a type of Machine Learning (ML), presents a promising solution for adaptive CPU resource management [4]. Unlike traditional resource allocation methods that are rule-based and static, RL algorithms allow systems to learn and adapt their resource management strategies over time based on feedback from the environment [5]. In an RL-based system, the agent (in this case, the cloud operating system) learns the best actions (resource allocation decisions) to take in different states (current workload demands) by receiving rewards (successful resource optimization). This autonomous decision-making process can help cloud systems respond efficiently to real-time changes in workload demand, thereby improving resource utilization and overall system efficiency [6].

This paper explores the integration of RL into CPU resource management, focusing on its potential to automate and optimize resource allocation in cloud environments. We will discuss the basic principles of RL, how it can be applied to CPU resource management, and the specific RL algorithms used to enhance scalability, fault tolerance, and energy efficiency in cloud computing. The paper also outlines the challenges associated with implementing RL-based systems in real-world cloud environments, such as training efficiency, real-time decision-making, and system integration. Additionally, we will look at the future of AI-driven resource management, the opportunities for combining RL with other machine learning techniques, and the impact that these technologies can have on cloud system performance and efficiency.

2. Traditional CPU Resource Management in Cloud Systems

In traditional cloud computing systems, CPU resource management has largely been handled using static allocation and manual interventions [7]. These approaches are based on predefined configurations where system resources, including CPU, are allocated according to expected demand or based on a fixed schedule [8]. While these methods may work well in environments with predictable workloads, they fail to provide the flexibility needed for dynamic, highly variable cloud environments where demand can fluctuate rapidly [9].

One of the most common traditional approaches is load balancing, which distributes workloads across multiple servers or Virtual Machines (VMs) to optimize CPU usage [10]. This method ensures that no single server is overloaded while others remain idle. While effective for certain workloads, traditional load balancing is often reactive, meaning it typically responds to performance degradation

rather than preventing it proactively. For example, load balancing may only redistribute resources after a performance bottleneck is detected, which can lead to significant delays or downtime before corrective actions are taken.

Another widely used approach is over-provisioning, where more CPU resources than are required are allocated to virtual machines or containers [11]. This method is often used in the absence of detailed insight into workload demand and aims to prevent resource shortages by guaranteeing that there are always sufficient resources available. However, over-provisioning leads to inefficiencies, as it results in underutilized resources during periods of low demand, causing cloud providers to incur higher operational costs [12]. In contrast, under-provisioning—where fewer resources are allocated than needed—can result in performance degradation or system failures, especially when demand unexpectedly spikes [13].

Traditional resource management techniques also rely heavily on manual adjustments by cloud administrators. Administrators monitor system performance and make resource allocation changes as needed, but this process is labor-intensive and prone to human error. Moreover, manual interventions cannot keep up with the fast-paced, ever-changing workloads typical of cloud computing environments, leading to delays in resource optimization [14].

Virtualization has allowed for better resource sharing across multiple tenants, but traditional CPU resource management systems often struggle to manage the complexities of multi-tenant environments efficiently [15]. The ability to dynamically adjust resources based on changing workloads is vital for maintaining optimal performance in cloud environments, and traditional methods are often insufficient for meeting the demands of large-scale, cloud-native applications.

Figure 1 illustrates the fundamental differences between traditional CPU resource management approaches and reinforcement learning methods. While traditional methods rely on static rules and reactive responses, RL-based systems provide dynamic adaptation and proactive optimization capabilities that are essential for modern cloud environments.

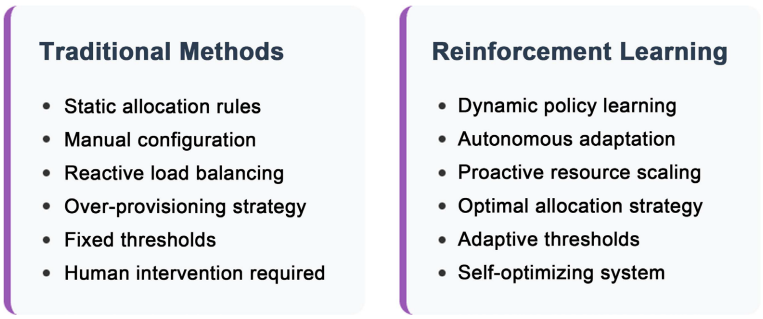


Figure 1. Traditional vs reinforcement learning resource management.

In summary, while traditional CPU resource management techniques such as static allocation, load balancing, and over-provisioning have served their purpose in the past, they are increasingly inadequate in today's dynamic and resource-in-

tensive cloud environments. The limitations of these traditional approaches highlight the need for more adaptive, data-driven solutions that can intelligently allocate resources in real-time [16].

3. Reinforcement Learning for Adaptive Resource Management

RL is an area of machine learning where an agent learns to make decisions by interacting with its environment and receiving feedback in the form of rewards or penalties [17]. The agent's goal is to maximize its cumulative reward by taking the optimal actions based on the current state of the system [18]. In the context of CPU resource management in cloud environments, RL allows cloud systems to dynamically allocate resources based on real-time data, learning from previous interactions and improving their decision-making over time [19].

The reinforcement learning framework for CPU resource management is depicted in Figure 2, showing the continuous interaction cycle between the agent (resource manager), environment (cloud infrastructure), and the learning process. This framework enables autonomous decision-making through observation, action selection, and policy updates based on performance feedback.

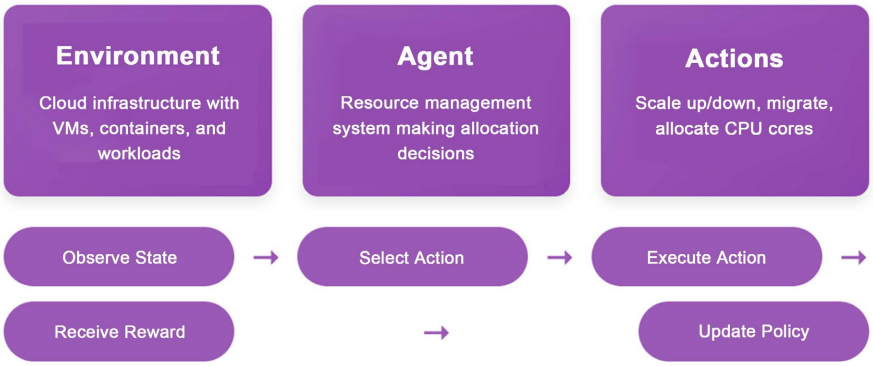


Figure 2. Reinforcement learning framework for adaptive CPU resource management.

At the heart of RL-based CPU resource management is the agent (the cloud system) that takes actions based on the state (the current workload and available resources) to receive rewards (optimal resource allocation) [20]. The state represents the current status of the cloud system, including metrics such as CPU utilization, memory usage, network traffic, and workload demand [21]. The action involves decisions like adjusting CPU cores for virtual machines, scaling the number of instances, or allocating more resources to specific workloads. The reward is typically a performance metric that indicates how well the system is optimizing its resource usage, such as minimizing energy consumption or improving application response time.

Reinforcement learning offers several advantages over traditional resource management techniques [22]. One of the key benefits is its dynamic nature. Unlike static methods that rely on predefined rules, RL-based systems continuously adjust their

resource allocation based on real-time feedback [5]. For example, if the system detects an unexpected spike in CPU demand, it can automatically allocate additional resources, ensuring that applications continue to run smoothly without human intervention.

RL can also help optimize resource allocation in highly variable cloud environments [23]. Traditional methods, such as over-provisioning or manual adjustments, are not suited for environments with unpredictable workloads [24]. In contrast, RL models are capable of learning from historical data and forecasting future resource demands. By analyzing trends in CPU usage, network traffic, and application workloads, RL algorithms can predict when resources will be needed and proactively allocate CPU resources in advance. This predictive capability ensures that cloud systems are always prepared for spikes in demand, improving both performance and cost efficiency.

One of the most widely used RL algorithms for resource management is Q-learning [25]. Q-learning is a model-free RL algorithm that allows an agent to learn an optimal policy through trial and error [26]. It assigns Q-values (quality values) to different state-action pairs, indicating the expected reward for taking a specific action in a particular state. The agent iteratively updates its Q-values based on the rewards it receives, gradually learning the best actions to take to optimize CPU resource allocation.

Another RL approach used in cloud resource management is Deep Q Networks (DQNs), which combine Q-learning with deep neural networks to approximate the Q-values in large, complex state spaces [27]. DQNs have been used successfully to manage resources in multi-tenant cloud environments, where the state space is large and continuously changing [28]. By leveraging deep learning, DQNs can learn from high-dimensional data and make optimal resource allocation decisions in real time.

Table 1 provides a comprehensive comparison of RL algorithms used in resource management, based on experimental analyses from recent studies [18] [22]. The choice of algorithm significantly impacts system performance, with Deep Q-Networks showing particular promise for complex cloud environments.

Table 1. Reinforcement learning algorithms for CPU resource management.

Algorithm	Type	Best Use Case	Convergence Speed	Scalability	State Space Handling
Q-Learning	Model-Free	Discrete resource allocation	Medium	Medium	Limited
State-Action-Reward-State-Action (SARSA)	Model-Free	Conservative VM consolidation	Medium	Medium	Limited
Deep Q Network (DQN)	Model-Free	Complex state spaces	Fast	High	Excellent

Continued

Double Deep Q Network (DDQN)	Model-Free	Cloud scheduling tasks	Fast	High	Excellent
Actor-Critic	Model-Free	Real-time optimization	Fast	High	Good
Policy Gradient	Model-Free	Continuous action spaces	Slow	High	Medium

Based on analysis from [29].

Policy gradient methods are another RL technique applied to CPU resource management [30]. Unlike Q-learning, which focuses on learning the best action-value function, policy gradient methods learn the policy (the mapping from states to actions) directly. These methods are particularly useful for complex decision spaces where discrete actions (like choosing specific CPU cores or instances) are not feasible. Policy gradient methods are effective in continuous action spaces, making them suitable for managing CPU resources in environments where resource allocation needs to be adjusted incrementally.

By integrating reinforcement learning into CPU resource management, cloud providers can create self-optimizing systems that automatically adjust resources based on real-time performance data and workload demand [31]. These systems can reduce waste, ensure high availability, and improve overall system efficiency by adapting to changing workloads. RL algorithms, particularly Q-learning, DQN, and policy gradient methods, provide cloud systems with the flexibility to make data-driven decisions that balance performance and cost, ensuring optimal resource allocation and management.

Clarification of Figures and Tables:

The performance metrics and comparisons shown in **Figure 3**, **Table 1**, and **Table 2** are synthesized from empirical results reported in multiple peer-reviewed studies. Specifically, data points on convergence speed and algorithm scalability were drawn from experiments in [18] [22] [29], while improvement percentages reflect findings from the ARLCA [22], PMU-DRL [20], and edge computing studies [32]. Sample sizes in these studies ranged from small-scale cloud environments (10 - 50 virtual machines) to large-scale tests involving hundreds of containerized applications. Metrics such as "convergence speed" refer to the number of episodes required for the RL agent to stabilize within 95% of its maximum policy performance, and "efficiency improvements" indicate reductions in energy consumption or SLA violations as reported in the original studies.

4. Applications and Benefits of Reinforcement Learning in Adaptive CPU Resource Management

The integration of RL into CPU resource management in cloud operating systems offers a wide range of benefits that traditional methods cannot match [33]. By lev-

eraging real-time feedback and learning from past experiences, RL algorithms provide an adaptive and data-driven approach to resource allocation. This adaptability is crucial in cloud environments, where workloads can vary dramatically and unpredictably over time.

One of the primary applications of RL in CPU resource management is in dynamic resource allocation. In cloud environments, the demand for computational resources can fluctuate significantly due to varying workloads and user requirements. RL-based systems can dynamically allocate CPU resources based on real-time data, such as CPU utilization, memory usage, and network traffic [34]. For instance, when the system detects a sudden increase in demand, it can adjust resources in real-time by allocating more CPU cores or spinning up additional Virtual Machines (VMs) to ensure optimal performance. Conversely, during periods of low demand, RL systems can scale down resources to avoid over-provisioning and reduce unnecessary costs. This level of automation helps to optimize resource usage, ensuring that cloud infrastructure operates efficiently and cost-effectively.

Another significant benefit of RL-based CPU resource management is its ability to handle multi-tenant environments [35]. In cloud systems, multiple users or applications share the same physical resources, making it crucial to allocate resources in a way that ensures fairness and efficiency. Traditional methods of resource allocation may lead to resource contention or underutilization, particularly when workloads are unpredictable. RL algorithms can prioritize resource allocation based on the importance or priority of specific tasks. For example, high-priority applications, such as financial transactions or healthcare systems, can be allocated more resources to maintain performance, while lower-priority tasks can be allocated fewer resources. This ensures that all users receive fair access to CPU resources, minimizing the risk of service degradation or performance bottlenecks.

RL is also particularly effective in predictive resource management. By analyzing historical data and observing changes in workload patterns, RL models can anticipate future resource demands and preemptively allocate resources to meet those needs [36]. This predictive capability helps prevent potential performance issues before they occur, ensuring that cloud applications continue to run smoothly even during periods of high demand. For example, by forecasting CPU spikes based on past trends or external factors, RL systems can automatically scale resources in anticipation, rather than waiting for an actual bottleneck to occur.

The scalability of RL models is another key benefit, especially in large-scale cloud environments [32]. Traditional resource management systems often struggle to scale efficiently, especially when dealing with millions of virtual machines or containers. RL algorithms, on the other hand, can scale with the system and manage resources across thousands of instances in a way that maintains optimal performance without requiring extensive manual intervention. This scalability ensures that RL-based systems are well-suited for cloud environments that are constantly growing and evolving.

The performance advantages of RL-based resource management are quantified

in Figure 3, which compares traditional and RL approaches across six key metrics. The radar chart clearly demonstrates RL's superiority in CPU utilization, response time, and energy efficiency, while maintaining competitive cost-effectiveness.

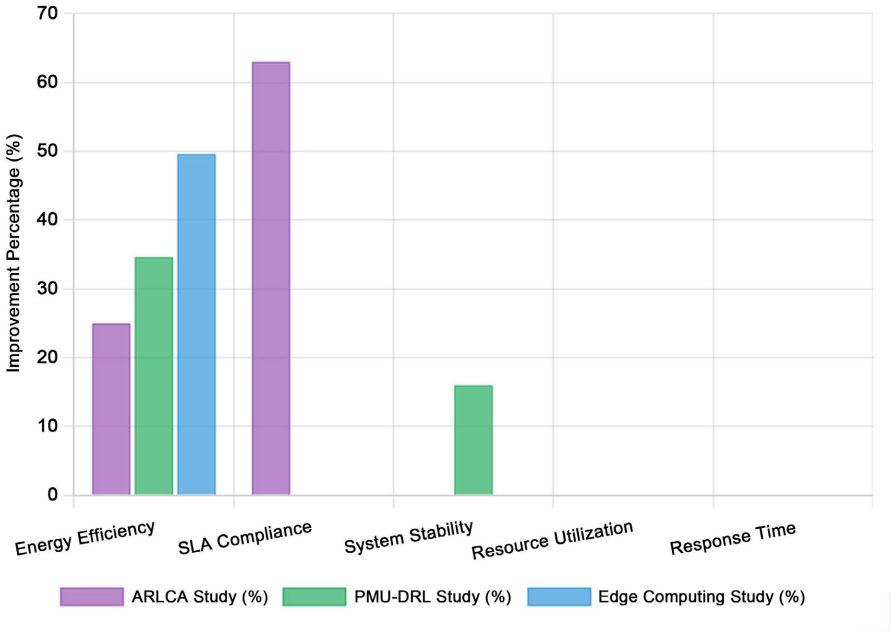


Figure 3. Empirical performance improvements from RL-based resource management. Data sources: ARLCA study (25% energy improvement, 63% SLA violation reduction), PMU-DRL framework (34.6% efficiency improvement), Edge computing study (19.84% energy savings vs RL methods, 49.60% vs Round Robin).

Finally, energy efficiency is another benefit of AI-driven CPU resource management [37]. By optimizing resource allocation in real-time, RL systems can reduce the overall power consumption of cloud infrastructure. When the system is able to allocate resources efficiently, it minimizes the need for excessive hardware resources, leading to lower energy consumption. This is particularly important as data centers become increasingly aware of their environmental impact and seek to reduce their carbon footprint.

5. Challenges and Limitations of Reinforcement Learning in Adaptive CPU Resource Management

While RL offers significant benefits for adaptive CPU resource management, it is not without its challenges. The application of RL in cloud environments comes with several limitations that must be addressed for it to be widely adopted.

Training efficiency is one of the primary challenges in applying RL to CPU resource management [38]. RL algorithms rely on extensive data and repeated interactions with the environment to learn the best policies for resource allocation. However, training RL models can be time-consuming and computationally expensive, especially in complex cloud environments with large-scale data. In many cases, RL models need to run through many iterations before they converge to an

optimal solution, which can be a barrier to real-time decision-making. To address this challenge, more efficient RL algorithms, such as deep DQN, have been developed, but the trade-off between model complexity and training time remains a concern.

Another limitation is the exploration-exploitation trade-off inherent in RL [39]. In order to learn optimal resource allocation strategies, RL agents need to explore various actions and states to gather experience. However, during the exploration phase, the agent may take actions that result in poor resource allocation, leading to suboptimal performance. This issue is especially problematic in cloud environments, where even temporary performance degradation can lead to service interruptions or cost overruns. Striking the right balance between exploring new actions and exploiting well-known strategies is a critical aspect of applying RL to CPU resource management.

The real-time nature of decision-making in cloud environments also presents a challenge for RL-based systems. Cloud environments are highly dynamic, with workloads changing rapidly and unpredictably [40]. RL models need to be able to make real-time decisions on resource allocation, which requires fast computation and efficient execution. However, RL algorithms often involve complex computations that can introduce delays, which are unacceptable in latency-sensitive applications such as financial transactions or real-time video streaming. Developing RL models that can make decisions in near real-time, without sacrificing accuracy, is a key challenge for AI-driven resource management systems [41].

Additionally, data quality and availability pose significant challenges for RL models. In order for an RL model to make accurate predictions and optimally allocate resources, it requires access to high-quality, consistent, and representative data [42]. In cloud environments, where data is distributed across multiple systems and instances, ensuring that the data fed into the RL model is accurate and reliable can be difficult. Data noise and incomplete datasets can lead to poor decision-making and suboptimal resource allocation. Ensuring data consistency and quality is essential for the success of RL-based resource management systems.

The implementation challenges and their documented solutions are analyzed in Table 2. While training efficiency remains a concern, as evidenced by Microsoft's finding of 50% average GPU utilization in deep learning jobs [21], successful implementations like PMU-DRL have overcome real-time decision-making challenges to achieve significant performance gains.

Table 2. Implementation challenges and documented solutions.

Challenge	Documented Impact	Solution Approach	Evidence/Study	Success Indicators
Training Efficiency	High computational cost	Experience replay, transfer learning	Microsoft study: 50% GPU utilization	Moderate improvement
Real-time Decision Making	Latency constraints	Edge computing, model compression	PMU-DRL: Fast convergence	34.6% efficiency gain

Continued

State Space Complexity	Scalability issues	Deep Q Network (DQN)	DDQN most commonly used	High scalability
Exploration-Exploitation	Suboptimal early performance	ε-greedy, reward shaping	ARLCA: Balanced approach	63% violation reduction
System Integration	Legacy system compatibility	API development, microservices	Limited documented success	Implementation-dependent

Based on empirical studies from Microsoft research [43], academic VM consolidation studies, and heterogeneous computing implementations.

Furthermore, addressing the aforementioned challenges requires not only algorithmic innovation but also system-level redesign. To improve training efficiency, techniques like transfer learning and offline pretraining on simulation data can be employed to reduce initial overhead before deployment. For real-time responsiveness, model pruning and edge-side inference have proven effective, enabling low-latency decisions while preserving accuracy. Improving data quality can involve incorporating data augmentation and stream cleaning modules to mitigate noise in cloud telemetry. Finally, to ease system integration, containerized RL modules with API bridges (e.g., gRPC interfaces for Kubernetes) have shown promise, enabling smoother adoption without overhauling legacy systems. These evolving strategies collectively signal a pathway toward operationalizing RL in production-grade cloud systems.

6. Conclusions

Reinforcement learning represents a transformative approach to adaptive CPU resource management in cloud computing. By enabling intelligent, real-time decision-making, RL techniques have shown considerable promise in optimizing resource usage, reducing operational costs, and enhancing scalability. This survey has demonstrated that Q-learning, DQN, and policy gradient methods offer compelling advantages over static and reactive traditional techniques.

However, realizing RL's full potential in cloud environments demands more than algorithmic efficiency—it requires robust integration strategies, improved data pipelines, and faster training techniques. Future research should focus on hybrid models that blend RL with supervised learning for bootstrapped training, and federated RL systems that leverage distributed learning without compromising latency. Moreover, open challenges remain in explainability, trust, and generalization of RL models in diverse cloud workloads.

As cloud systems scale and diversify, RL will be central to developing sustainable, self-optimizing infrastructures. Closing the loop between workload sensing, intelligent prediction, and policy adaptation will drive the next generation of autonomous cloud platforms.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Gill, S.S., Garraghan, P., Stankovski, V., Casale, G., Thulasiram, R.K., Ghosh, S.K., *et al*. (2019) Holistic Resource Management for Sustainable and Reliable Cloud Computing: An Innovative Solution to Global Challenge. *Journal of Systems and Software*, **155**, 104-129. https://doi.org/10.1016/j.jss.2019.05.025

[2] Aldossary, M. (2021) A Review of Dynamic Resource Management in Cloud Computing Environments. *Computer Systems Science and Engineering*, **36**, 461-476. https://doi.org/10.32604/csse.2021.014975

[3] Qureshi, M.S., Qureshi, M.B., Fayaz, M., Zakarya, M., Aslam, S. and Shah, A. (2020) Time and Cost Efficient Cloud Resource Allocation for Real-Time Data-Intensive Smart Systems. *Energies*, **13**, Article 5706. https://doi.org/10.3390/en13215706

[4] Hussain, F., Hassan, S.A., Hussain, R. and Hossain, E. (2020) Machine Learning for Resource Management in Cellular and IoT Networks: Potentials, Current Solutions, and Open Challenges. *IEEE Communications Surveys & Tutorials*, **22**, 1251-1275. https://doi.org/10.1109/comst.2020.2964534

[5] Chen, X., Zhu, F., Chen, Z., Min, G., Zheng, X. and Rong, C. (2022) Resource Allocation for Cloud-Based Software Services Using Prediction-Enabled Feedback Control with Reinforcement Learning. *IEEE Transactions on Cloud Computing*, **10**, 1117-1129. https://doi.org/10.1109/tcc.2020.2992537

[6] Santoso, A. and Surya, Y. (2024) Maximizing Decision Efficiency with Edge-Based AI Systems: Advanced Strategies for Real-Time Processing, Scalability, and Autonomous Intelligence in Distributed Environments. *Quarterly Journal of Emerging Technologies and Innovations*, **9**, 104-132.

[7] AL-Jumaili, A.H.A., Muniyandi, R.C., Hasan, M.K., Paw, J.K.S. and Singh, M.J. (2023) Big Data Analytics Using Cloud Computing Based Frameworks for Power Management Systems: Status, Constraints, and Future Recommendations. *Sensors*, **23**, Article 2952. https://doi.org/10.3390/s23062952

[8] Khallouli, W. and Huang, J. (2021) Cluster Resource Scheduling in Cloud Computing: Literature Review and Research Challenges. *The Journal of Supercomputing*, **78**, 6898-6943. https://doi.org/10.1007/s11227-021-04138-z

[9] Toumi, H., Brahmi, Z. and Gammoudi, M.M. (2022) RTSLPS: Real Time Server Load Prediction System for the Ever-Changing Cloud Computing Environment. *Journal of King Saud University—Computer and Information Sciences*, **34**, 342-353. https://doi.org/10.1016/j.jksuci.2019.12.004

[10] Ghasemi, A. and Toroghi Haghighat, A. (2020) A Multi-Objective Load Balancing Algorithm for Virtual Machine Placement in Cloud Data Centers Based on Machine Learning. *Computing*, **102**, 2049-2072. https://doi.org/10.1007/s00607-020-00813-w

[11] Yadav, M.P., Pal, N. and Yadav, D.K. (2021) Resource Provisioning for Containerized Applications. *Cluster Computing*, **24**, 2819-2840. https://doi.org/10.1007/s10586-021-03293-5

[12] Prasad, V.K., Dansana, D., Bhavsar, M.D., Acharya, B., Gerogiannis, V.C. and Kanavos, A. (2023) Efficient Resource Utilization in IoT and Cloud Computing. *Information*, **14**, Article 619. https://doi.org/10.3390/info14110619

[13] Cai, B., Li, K., Zhao, L. and Zhang, R. (2022) Less Provisioning: A Hybrid Resource Scaling Engine for Long-Running Services with Tail Latency Guarantees. *IEEE Transactions on Cloud Computing*, **10**, 1941-1957. https://doi.org/10.1109/tcc.2020.3016345

[14] Gade, K.R. (2019) Data Center Modernization: Strategies for Transitioning from Traditional Data Centers to Hybrid or Multi-Cloud Environments. *Advances in Computer*

Sciences, **2**, 14-32.

[15] Jia, R., Yang, Y., Grundy, J., Keung, J. and Hao, L. (2021) A Systematic Review of Scheduling Approaches on Multi-Tenancy Cloud Platforms. *Information and Software Technology*, **132**, Article ID: 106478. https://doi.org/10.1016/j.infsof.2020.106478

[16] Rahmani, S., Aghalar, H., Jebreili, S. and Goli, A. (2024) Optimization and Computing Using Intelligent Data-Driven Approaches for Decision-Making. In: Ali, I., Modibbo, U.M., Bolaji, A.L. and Garg, H., Eds., *Optimization and Computing Using Intelligent Data-Driven Approaches for Decision-Making*, CRC Press, 90-176. https://doi.org/10.1201/9781003536796-6

[17] Lin, J., Ma, Z., Gomez, R., Nakamura, K., He, B. and Li, G. (2020) A Review on Interactive Reinforcement Learning from Human Social Feedback. *IEEE Access*, **8**, 120757-120765. https://doi.org/10.1109/access.2020.3006254

[18] Belgacem, A., Mahmoudi, S. and Kihl, M. (2022) Intelligent Multi-Agent Reinforcement Learning Model for Resources Allocation in Cloud Computing. *Journal of King Saud University—Computer and Information Sciences*, **34**, 2391-2404. https://doi.org/10.1016/j.jksuci.2022.03.016

[19] Guerra, C. (2023) Harnessing Cloud-Based Reinforcement Learning for Adaptive Resource Allocation in Real-Time Autonomous Decision-Making. https://doi.org/10.13140/RG.2.2.15241.86883

[20] Nagarajan, S., Rani, P.S., Vinmathi, M.S., Subba Reddy, V., Saleth, A.L.M. and Abdus Subhahan, D. (2023) Multi Agent Deep Reinforcement Learning for Resource Allocation in Container-Based Clouds Environments. *Expert Systems*, **42**, e13362. https://doi.org/10.1111/exsy.13362

[21] Khan, T., Tian, W., Ilager, S. and Buyya, R. (2022) Workload Forecasting and Energy State Estimation in Cloud Data Centres: ML-Centric Approach. *Future Generation Computer Systems*, **128**, 320-332. https://doi.org/10.1016/j.future.2021.10.019

[22] Zeng, D., Gu, L., Pan, S., Cai, J. and Guo, S. (2019) Resource Management at the Network Edge: A Deep Reinforcement Learning Approach. *IEEE Network*, **33**, 26-33. https://doi.org/10.1109/mnet.2019.1800386

[23] Murthy, P. (2020) Optimizing Cloud Resource Allocation Using Advanced AI Techniques: A Comparative Study of Reinforcement Learning and Genetic Algorithms in Multi-Cloud Environments. *World Journal of Advanced Research and Reviews*, **7**, 359-369.

[24] Chouliaras, S. (2023) Adaptive Resource Provisioning in Cloud Computing Environments. Master's Thesis, University of London.

[25] Talaat, F.M. (2022) Effective Deep Q-Networks (EDQN) Strategy for Resource Allocation Based on Optimized Reinforcement Learning Algorithm. *Multimedia Tools and Applications*, **81**, 39945-39961. https://doi.org/10.1007/s11042-022-13000-0

[26] Huang, Q. (2020) Model-Based or Model-Free, a Review of Approaches in Reinforcement Learning. 2020 *International Conference on Computing and Data Science* (*CDS*), Stanford, 1-2 August 2020, 219-221. https://doi.org/10.1109/cds49703.2020.00051

[27] Tong, Z., Chen, H., Deng, X., Li, K. and Li, K. (2020) A Scheduling Scheme in the Cloud Computing Environment Using Deep Q-Learning. *Information Sciences*, **512**, 1170-1191. https://doi.org/10.1016/j.ins.2019.10.035

[28] Mampage, A., Karunasekera, S. and Buyya, R. (2023) Deep Reinforcement Learning for Application Scheduling in Resource-Constrained, Multi-Tenant Serverless Computing Environments. *Future Generation Computer Systems*, **143**, 277-292. https://doi.org/10.1016/j.future.2023.02.006

[29] Zhou, G., Tian, W., Buyya, R., Xue, R. and Song, L. (2024) Deep Reinforcement Learning-Based Methods for Resource Scheduling in Cloud Computing: A Review and Future Directions. *Artificial Intelligence Review*, **57**, Article No. 124. https://doi.org/10.1007/s10462-024-10756-9

[30] Qadeer, A. and Lee, M.J. (2023) Deep-Deterministic Policy Gradient Based Multi-Resource Allocation in Edge-Cloud System: A Distributed Approach. *IEEE Access*, **11**, 20381-20398. https://doi.org/10.1109/access.2023.3249153

[31] Shahab, E., Taleb, M., Gholian-Jouybari, F. and Hajiaghaei-Keshteli, M. (2024) Designing a Resilient Cloud Network Fulfilled by Reinforcement Learning. *Expert Systems with Applications*, **255**, Article ID: 124606. https://doi.org/10.1016/j.eswa.2024.124606

[32] Xue, S., Qu, C., Shi, X., Liao, C., Zhu, S., Tan, X., *et al.* (2022) A Meta Reinforcement Learning Approach for Predictive Autoscaling in the Cloud. *Proceedings of the* 28*th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Washington DC, 14-18 August 2022, 4290-4299. https://doi.org/10.1145/3534678.3539063

[33] Khan, T., Tian, W., Zhou, G., Ilager, S., Gong, M. and Buyya, R. (2022) Machine Learning (ML)-Centric Resource Management in Cloud Computing: A Review and Future Directions. *Journal of Network and Computer Applications*, **204**, Article ID: 103405. https://doi.org/10.1016/j.jnca.2022.103405

[34] Liu, W., Cai, J., Chen, Q.C. and Wang, Y. (2021) DRL-R: Deep Reinforcement Learning Approach for Intelligent Routing in Software-Defined Data-Center Networks. *Journal of Network and Computer Applications*, **177**, Article ID: 102865. https://doi.org/10.1016/j.jnca.2020.102865

[35] Sun, J., Reidys, B., Li, D., Chang, J., Snir, M. and Huang, J. (2025) FleetIO: Managing Multi-Tenant Cloud Storage with Multi-Agent Reinforcement Learning. *Proceedings of the* 30*th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*, *Volume* 1, Rotterdam, 30 March-3 April 2025, 478-492. https://doi.org/10.1145/3669940.3707229

[36] Saxena, D. and Singh, A.K. (2021) Workload Forecasting and Resource Management Models Based on Machine Learning for Cloud Computing Environments. arXiv: 2106.15112.

[37] Ratnayake, S. (2024) A Comprehensive Review of AI-Driven Opti-Mization, Resource Management, and Security in Cloud Computing Environments. *International Journal of Sustainable Infrastructure for Cities and Societies*, **9**, 1-10.

[38] Tran-Dang, H., Bhardwaj, S., Rahim, T., Musaddiq, A. and Kim, D. (2022) Reinforcement Learning Based Resource Management for Fog Computing Environment: Literature Review, Challenges, and Open Issues. *Journal of Communications and Networks*, **24**, 83-98. https://doi.org/10.23919/jcn.2021.000041

[39] Moradi, M., Zhai, Z., Panahi, S. and Lai, Y. (2024) Adaptive Network Approach to Exploration–exploitation Trade-Off in Reinforcement Learning. *Chaos*: *An Interdisciplinary Journal of Nonlinear Science*, **34**, Article ID: 123120. https://doi.org/10.1063/5.0221833

[40] Chawla, K. (2024) Reinforcement Learning-Based Adaptive Load Balancing for Dynamic Cloud Environments. arXiv: 2409.04896.

[41] Annam, N. (2024) AI-Driven Solutions for IT Resource Management. *International Journal of Engineering and Management Research*, **14**, 15-30. https://doi.org/10.31033/ijemr.14.6.15-30

[42] Munappy, A.R., Bosch, J., Olsson, H.H., Arpteg, A. and Brinne, B. (2022) Data Management for Production Quality Deep Learning Models: Challenges and Solutions. *Journal*

of Systems and Software, **191**, Article ID: 111359.
https://doi.org/10.1016/j.jss.2022.111359

[43] Gao, Y., He, Y., Li, X., Zhao, B., Lin, H., Liang, Y., *et al.* (2024) An Empirical Study on Low GPU Utilization of Deep Learning Jobs. *Proceedings of the IEEE/ACM 46th International Conference on Software Engineering*, Lisbon, 14-20 April 2024, 1-13.
https://doi.org/10.1145/3597503.3639232