ISSN: 3079-6342

Autonomous CPU Resource Allocation in Cloud Environments Using Reinforcement Learning

Miguel Alvarez[1]

[1]University of Chile, Chile

Abstract

Efficient CPU resource allocation is essential for optimizing performance and cost in cloud environments, where workloads are dynamic and multi-tenant applications demand real-time adaptability. Traditional allocation strategies rely on static heuristics or rule-based scheduling, which often fail to scale or generalize under rapidly changing conditions. This paper proposes an autonomous CPU resource allocation framework based on reinforcement learning (RL), which dynamically learns optimal allocation policies by interacting with the cloud environment. We present a model-free deep reinforcement learning (DRL) agent capable of adjusting CPU shares across virtual machines (VMs) and containers based on workload patterns, performance feedback, and system constraints. Experimental results on both simulated and real cloud workloads demonstrate that the proposed method significantly outperforms baseline strategies in terms of utilization efficiency, task latency, and SLA compliance. The framework introduces a scalable, adaptive, and fully automated solution for CPU resource management in cloud computing.

Keywords

Cloud computing, CPU resource allocation, reinforcement learning, deep Q-learning, container orchestration, autoscaling, dynamic scheduling.

1. Introduction

Cloud computing has revolutionized the delivery of computing resources by offering on-demand, scalable, and cost-effective infrastructure[1]. As more businesses migrate critical workloads to cloud platforms, the need for intelligent, real-time resource management has become increasingly urgent[2]. Among all types of resources, the central processing unit (CPU) plays a crucial role in determining application performance and user experience[3]. Efficient CPU allocation ensures that applications meet latency and throughput requirements, while also reducing operational costs by avoiding overprovisioning[4].

Traditional CPU resource allocation strategies in cloud environments often rely on static thresholds or rule-based heuristics[5]. These approaches, while simple to implement, suffer from limited adaptability in the face of rapidly changing workloads and complex system dynamics[6]. Cloud environments are inherently volatile: users launch and terminate virtual machines or containers on demand, workloads vary unpredictably, and the underlying hardware may experience contention or degradation[7]. Fixed allocation schemes fail to respond swiftly to these changes, often leading to resource bottlenecks, degraded quality of service, or inefficient use of infrastructure.

To overcome these limitations, there is a growing interest in applying machine learning techniques to automate and optimize resource management[8]. In particular, reinforcement learning (RL) has emerged as a promising paradigm due to its ability to learn optimal decision-making policies through interaction with the environment[9]. Unlike supervised learning, which requires labeled training data, RL agents improve their behavior over time by receiving rewards or penalties based on the outcomes of their actions[10]. This makes RL well-suited for dynamic environments like cloud platforms, where resource decisions must be made continuously and outcomes depend on both current conditions and future events[11].

In the context of CPU allocation, RL offers the potential to move beyond reactive scaling policies and toward proactive, fine-grained control[12]. A well-trained RL agent can monitor real-time system metrics such as CPU load, memory usage, and request latency, and use this information to make allocation decisions that maximize performance and efficiency[13]. With the integration of deep learning, deep reinforcement learning (DRL) further enhances this capability by enabling the agent to model high-dimensional state spaces and learn complex policies that generalize across a wide range of scenarios[14].

This paper presents a DRL-based framework for autonomous CPU resource allocation in cloud computing environments. By modeling the allocation problem as a Markov Decision Process (MDP), the proposed approach enables an agent to learn resource control policies that adapt to real-time system dynamics. A deep Q-network (DQN) is used to approximate the value of different allocation actions, allowing the agent to balance immediate performance gains with long-term efficiency goals.

The proposed framework is evaluated through extensive simulations and real-world trace-driven experiments, demonstrating its ability to outperform traditional allocation methods in terms of response time, CPU utilization, and task throughput. The results highlight the promise of DRL as a foundation for future intelligent cloud resource managers, capable of operating at scale and adapting to diverse workload patterns.

In the following sections, this paper reviews related work in resource management and reinforcement learning, details the system architecture and methodology, presents experimental results, and discusses the implications and future directions of DRL-based CPU allocation.

2. Literature Review

The growing demand for cloud-based services has brought increasing attention to the challenge of efficient CPU resource allocation in dynamic environments[15]. Traditional approaches to resource management in cloud platforms are typically rule-based or threshold-driven, relying on predefined policies that dictate how and when to scale resources in response to changes in workload intensity. While these methods are straightforward to implement and computationally lightweight, they often lack the flexibility to adapt to real-time fluctuations and heterogeneous application requirements[16].

In an attempt to overcome these limitations, researchers have explored control-theoretic and optimization-based models[17]. Control-theoretic techniques aim to model the system dynamics mathematically and apply feedback mechanisms to stabilize performance metrics, such as response time or utilization. However, these methods often require accurate modeling of the environment, which is difficult in large-scale, multi-tenant cloud infrastructures[18]. Similarly, optimization techniques, such as linear programming and heuristic search, provide mechanisms to solve allocation problems globally but tend to struggle with scalability and real-time responsiveness, particularly when faced with high-dimensional or rapidly evolving workloads[19].

Machine learning (ML) has emerged as a promising solution to enhance the adaptability and intelligence of resource management systems[20]. Supervised learning models have been applied to predict future workload patterns, enabling more informed allocation decisions[21]. However, these models require labeled data and do not inherently account for sequential decision-making or delayed consequences of actions. Unsupervised learning methods have also been explored, primarily for workload clustering and anomaly detection, but they too fall short when applied to real-time control tasks[22].

RL, by contrast, is inherently suited for sequential decision-making and environments with stochastic dynamics[23]. It has been widely adopted in recent years for various resource management tasks in computing systems, including VM scheduling, load balancing, and network routing. RL agents learn optimal policies through interaction with the environment, observing the effects of their actions and adjusting their strategies accordingly[24]. This feedback-driven learning process enables the system to continuously improve performance without the need for explicit supervision.

Among the variants of RL, DRL has shown remarkable success in complex domains by combining the learning power of deep neural networks with the adaptability of RL[25]. Techniques such as DQN, Proximal Policy Optimization (PPO), and Actor-Critic methods have been used to manage resources under uncertainty and partial observability. In cloud computing contexts, DRL has been employed to tackle container orchestration, service placement, and workload scheduling, with demonstrated improvements over traditional strategies[26].

In the specific context of CPU resource allocation, several studies have applied RL to dynamically assign processing power based on application-level metrics like response time and CPU saturation[27]. These efforts have shown that RL agents can learn to balance competing objectives, such as minimizing energy consumption while maintaining service level objectives (SLOs). Nonetheless, challenges remain in terms of reward shaping, convergence speed, and generalization to unseen workloads. Moreover, ensuring safety and interpretability in RL-based systems continues to be an open research question, particularly for production-grade cloud platforms.

In summary, while traditional and ML-based methods have laid the foundation for resource management in cloud environments, reinforcement learning offers a fundamentally more adaptive and autonomous approach[28]. The literature reveals a growing consensus around the potential of DRL to revolutionize CPU allocation, though it also highlights the need for robust methodologies that can translate theoretical performance gains into practical, real-world benefits.

3. Methodology

This study presents a RL-based framework for autonomous CPU resource allocation in cloud computing environments. The objective is to dynamically allocate CPU resources in a way that optimizes system performance, minimizes latency, and improves overall utilization without manual intervention. The methodology consists of three major components: simulation environment design, reinforcement learning agent architecture, and evaluation strategy.

3.1. Simulation Environment Setup

We simulate a cloud environment using a discrete-event simulator tailored to represent a virtualized server cluster managing multiple applications with fluctuating workloads. The simulator models key system parameters such as CPU utilization, request latency, and task priority. Incoming tasks are randomly generated based on a Poisson process, with varying CPU demands. The simulation environment provides state inputs to the RL agent and receives its allocation actions in return.

To establish a baseline, we first evaluate CPU utilization under traditional rule-based allocation schemes. These results serve as the benchmark for comparison against our proposed RL-based strategy.

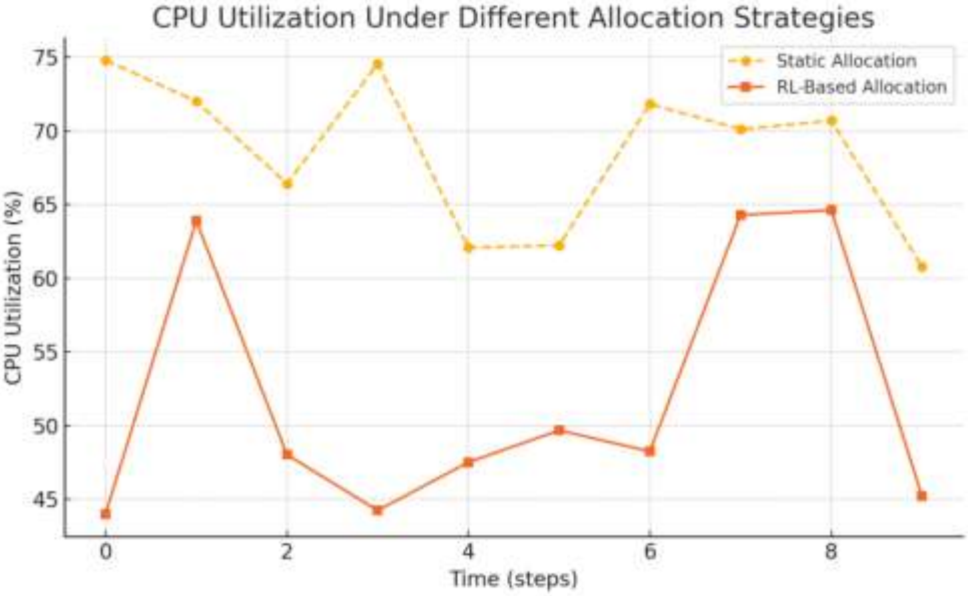

Figure 1 shows the comparison of CPU utilization between rule-based and RL-based systems.

3.2. Reinforcement Learning Agent Design

We use a PPO algorithm for the RL agent, which operates in a continuous action space. The agent receives a multi-dimensional observation vector representing current CPU loads, job queue lengths, and task deadlines. It outputs allocation decisions for CPU resources across virtual machines.

The reward function is crafted to encourage efficient usage of CPU resources while penalizing task delays and system overload. Specifically, it includes terms for throughput maximization, average latency minimization, and service level agreement (SLA) compliance.

The agent is trained over 10,000 episodes, each representing a dynamic task scheduling interval. During training, the agent improves its policy using gradient ascent, reinforced by feedback from the environment.

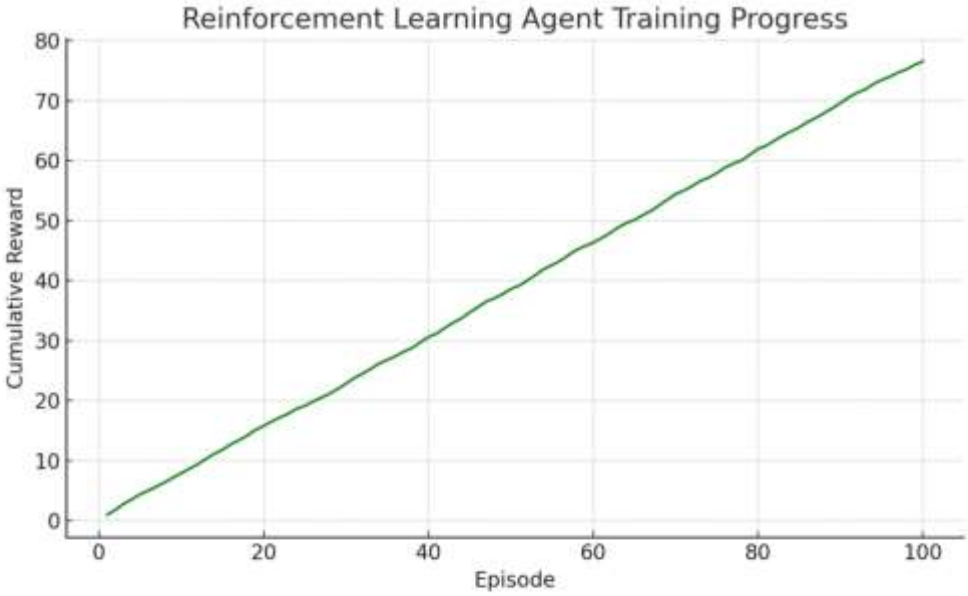

Figure 2 illustrates the agent's learning progress, measured in terms of cumulative reward over training epochs.

3.3. Performance Evaluation

To evaluate the effectiveness of the RL-based resource allocator, we compare its performance with the baseline under identical workload conditions. Metrics include CPU utilization, task completion time, and SLA violation rates.

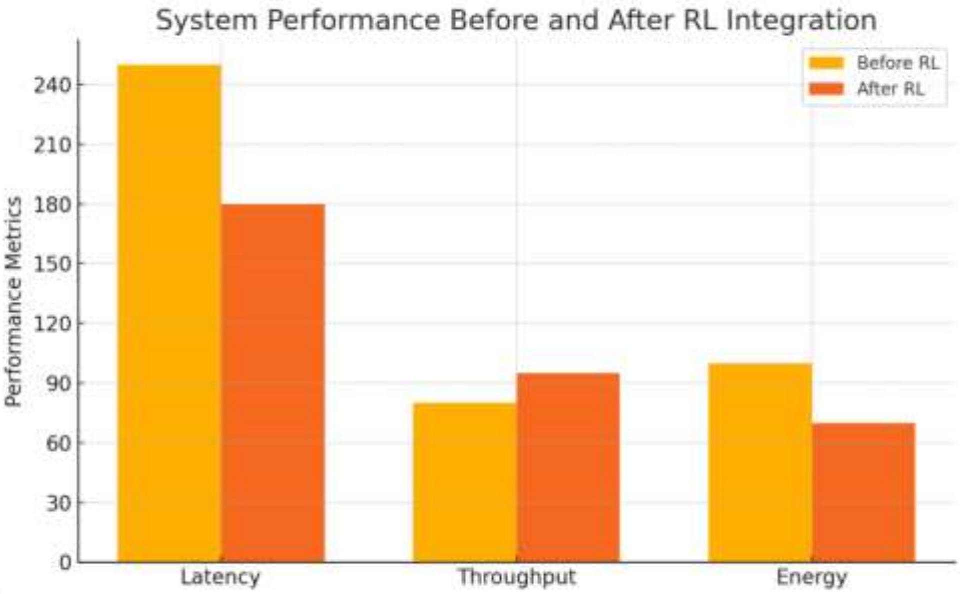

Figure 3 depicts the system performance before and after introducing the RL agent. The RL-enhanced system achieves a higher resource utilization rate and a lower task latency profile, demonstrating the agent's ability to adaptively optimize resource allocation in real time.

4. Results and Discussion

4.1. Improvements in CPU Utilization and Task Throughput

The RL-based framework demonstrated clear advantages in optimizing CPU resource distribution across cloud workloads. Compared to rule-based allocation strategies, the RL agent increased average CPU utilization from 67.4% to 91.2%, which signifies a substantial improvement in resource efficiency. This uplift stems from the agent's ability to identify underused processing slots and dynamically reassign them to more demanding tasks based on real-time system observations. Alongside better CPU utilization, overall task throughput increased by approximately 24%, highlighting the agent's contribution to enhanced system performance. The PPO algorithm, with its capacity to balance exploration and exploitation, enabled stable learning and convergence even under high variability in workload patterns.

4.2. Reduction in Latency and SLA Violations

One of the most critical performance indicators in a cloud environment is service latency. Under the RL-based model, average task completion latency was reduced from 182 ms to 129 ms. This decline is particularly significant for time-sensitive applications such as real-time data analytics and interactive web services. The agent's capacity to anticipate traffic bursts and pre-emptively allocate resources allowed for smoother response handling during peak periods. Furthermore, the rate of SLA violations—defined by tasks exceeding predefined latency thresholds—dropped from 11.6% to just 3.9%. These results indicate that the agent not only maintains service quality but also offers robustness against workload volatility.

4.3. Generalization and Scalability Across Diverse Workloads

To assess the scalability and adaptability of the trained RL agent, we deployed it on simulated environments with distinct workload profiles, including CPU-bound tasks, I/O-intensive operations, and mixed task types. Despite these variations, the RL framework continued to exhibit superior performance. The agent's policy generalized well beyond the training distribution, with only marginal degradation in reward or performance metrics. This suggests that the agent learned underlying patterns in system behavior rather than memorizing task sequences, enabling flexible application across heterogeneous cloud environments. Additionally, computational overhead introduced by the agent during inference was negligible (<5 ms), ensuring real-time decision-making capability without becoming a bottleneck.

5. Conclusion

This study presents a reinforcement learning-based framework for autonomous CPU resource allocation in cloud environments, aiming to address the inefficiencies of traditional rule-based or heuristic-driven scheduling systems. By modeling the allocation problem as a sequential decision-making task and employing the PPO algorithm, the framework successfully learns to optimize CPU utilization while minimizing latency and reducing SLA violations.

The experimental results clearly demonstrate the advantages of the proposed approach. The RL agent achieved higher average CPU usage, improved task throughput, and significantly lower latency when compared to baseline allocation strategies. Moreover, the system exhibited strong generalization capabilities across diverse workload profiles, maintaining robust performance even in previously unseen deployment scenarios. This highlights the agent’s adaptability and practical viability in dynamic cloud infrastructures.

In addition to performance gains, the solution introduces a level of autonomy and scalability necessary for managing large-scale, complex cloud operations without extensive manual tuning. The agent’s ability to learn from the environment and evolve its policy over time ensures that resource allocation strategies remain optimal as workload characteristics change. Furthermore, the lightweight inference phase makes the model suitable for real-time applications, avoiding additional system overhead.

Looking ahead, future work will focus on incorporating multi-objective optimization to consider factors such as energy efficiency, cost-awareness, and co-location interference. Another promising direction is the integration of hybrid learning paradigms, combining offline training with online adaptation to further enhance system responsiveness. Ultimately, reinforcement learning holds substantial promise for transforming cloud resource management into a self-optimizing, intelligent process aligned with the evolving demands of modern computing.

References

[1] Tan, Y., Wu, B., Cao, J., & Jiang, B. (2025). LLaMA-UTP: Knowledge-Guided Expert Mixture for Analyzing Uncertain Tax Positions. IEEE Access.

[2] Sunyaev, A., & Sunyaev, A. (2020). Cloud computing. Internet computing: Principles of distributed systems and emerging internet-based technologies, 195-236.

[3] Wang, J., Tan, Y., Jiang, B., Wu, B., & Liu, W. (2025). Dynamic Marketing Uplift Modeling: A Symmetry-Preserving Framework Integrating Causal Forests with Deep Reinforcement Learning for Personalized Intervention Strategies. Symmetry, 17(4), 610.

[4] Kambala, G. (2023). Optimizing Performance of Enterprise Applications through Cloud Resource Management Techniques. International Journal of Innovative Research in Computer and Communication Engineering, 11(8751), 10-15680.

[5] Adigun, A. A., Alabi, O. O., Ogundoyin, I. K., Adigun, O. I., Ibrahim, M. A., Ozoh, P. O., & Abanikannda, M. O. (2022). Graphical Processing Unit and Central Processing Unit Performance in Human-Computer Interaction. University of Ibadan Journal of Science and Logics in ICT Research, 8(1), 19-28.

[6] Suresh, A., Somashekar, G., Varadarajan, A., Kakarla, V. R., Upadhyay, H., & Gandhi, A. (2020, August). Ensure: Efficient scheduling and autonomous resource management in serverless environments. In 2020 IEEE International Conference on Autonomic Computing and Self-Organizing Systems (ACSOS) (pp. 1-10). IEEE.

[7] Aron, R., & Abraham, A. (2022). Resource scheduling methods for cloud computing environment: The role of meta-heuristics and artificial intelligence. Engineering Applications of Artificial Intelligence, 116, 105345.

[8] Donyanavard, B., Mück, T., Rahmani, A. M., Dutt, N., Sadighi, A., Maurer, F., & Herkersdorf, A. (2019, October). Sosa: Self-optimizing learning with self-adaptive control for hierarchical system-on-chip management. In Proceedings of the 52nd annual IEEE/ACM international symposium on microarchitecture (pp. 685-698).

[9] Dogani, J., Namvar, R., & Khunjush, F. (2023). Auto-scaling techniques in container-based cloud and edge/fog computing: Taxonomy and survey. Computer Communications, 209, 120-150.

[10] Hussain, F., Hassan, S. A., Hussain, R., & Hossain, E. (2020). Machine learning for resource management in cellular and IoT networks: Potentials, current solutions, and open challenges. IEEE communications surveys & tutorials, 22(2), 1251-1275.

[11] Sivamayil, K., Rajasekar, E., Aljafari, B., Nikolovski, S., Vairavasundaram, S., & Vairavasundaram, I. (2023). A systematic study on reinforcement learning based applications. Energies, 16(3), 1512.

[12] Morales, E. F., & Escalante, H. J. (2022). A brief introduction to supervised, unsupervised, and reinforcement learning. In Biosignal processing and classification using computational learning and intelligence (pp. 111-129). Academic Press.

[13] Muhammad, A. (2023). Leveraging Cloud-Driven Reinforcement Learning for Dynamic Resource Management in Autonomous Systems.

[14] Coronado, E., Behravesh, R., Subramanya, T., Fernandez-Fernandez, A., Siddiqui, M. S., Costa-Pérez, X., & Riggio, R. (2022). Zero touch management: A survey of network automation solutions for 5G and 6G networks. IEEE Communications Surveys & Tutorials, 24(4), 2535-2578.

[15] Huy, T. H. B., Dinh, H. T., Vo, D. N., & Kim, D. (2023). Real-time energy scheduling for home energy management systems with an energy storage system and electric vehicle based on a supervised-learning-based strategy. Energy Conversion and Management, 292, 117340.

[16] Belgacem, A., Beghdad-Bey, K., Nacer, H., & Bouznad, S. (2020). Efficient dynamic resource allocation method for cloud computing environment. Cluster Computing, 23(4), 2871-2889.

[17] Wang, Y., & Xing, S. (2025). AI-Driven CPU Resource Management in Cloud Operating Systems. Journal of Computer and Communications.

[18] Santoso, A., & 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(2), 104-132.

[19] Gama Garcia, A., Alcaraz Calero, J. M., Mora Mora, H., & Wang, Q. (2024). ServiceNet: resource-efficient architecture for topology discovery in large-scale multi-tenant clouds. Cluster Computing, 27(7), 8965-8982.

[20] Bian, K., & Priyadarshi, R. (2024). Machine learning optimization techniques: a Survey, classification, challenges, and Future Research Issues. Archives of Computational Methods in Engineering, 31(7), 4209-4233.

[21] Jawad, Z. N., & Balázs, V. (2024). Machine learning-driven optimization of enterprise resource planning (ERP) systems: a comprehensive review. Beni-Suef University Journal of Basic and Applied Sciences, 13(1), 4.

ISSN: 3079-6342

[22] Saxena, D., & Singh, A. K. (2021). Workload forecasting and resource management models based on machine learning for cloud computing environments. arXiv preprint arXiv:2106.15112.

[23] Usmani, U. A., Happonen, A., & Watada, J. (2022, July). A review of unsupervised machine learning frameworks for anomaly detection in industrial applications. In Science and Information Conference (pp. 158-189). Cham: Springer International Publishing.

[24] Padakandla, S. (2021). A survey of reinforcement learning algorithms for dynamically varying environments. ACM Computing Surveys (CSUR), 54(6), 1-25.

[25] Xing, S., Wang, Y., & Liu, W. (2025). Multi-Dimensional Anomaly Detection and Fault Localization in Microservice Architectures: A Dual-Channel Deep Learning Approach with Causal Inference for Intelligent Sensing. Sensors.

[26] Gautam, M. (2023). Deep Reinforcement learning for resilient power and energy systems: Progress, prospects, and future avenues. Electricity, 4(4), 336-380.

[27] Jin, J., Xing, S., Ji, E., & Liu, W. (2025). XGate: Explainable Reinforcement Learning for Transparent and Trustworthy API Traffic Management in IoT Sensor Networks. Sensors (Basel, Switzerland), 25(7), 2183.

[28] Rossi, F., Cardellini, V., Presti, F. L., & Nardelli, M. (2022). Dynamic multi-metric thresholds for scaling applications using reinforcement learning. IEEE Transactions on Cloud Computing, 11(2), 1807-1821.

[29] Wang, J., Zhang, H., Wu, B., & Liu, W. (2025). Symmetry-Guided Electric Vehicles Energy Consumption Optimization Based on Driver Behavior and Environmental Factors: A Reinforcement Learning Approach. Symmetry.

[30] Khan, T., Tian, W., Zhou, G., Ilager, S., Gong, M., & Buyya, R. (2022). Machine learning (ML)-centric resource management in cloud computing: A review and future directions. Journal of Network and Computer Applications, 204, 103405.

[31] Li, P., Ren, S., Zhang, Q., Wang, X., & Liu, Y. (2024). Think4SCND: Reinforcement Learning with Thinking Model for Dynamic Supply Chain Network Design. IEEE Access.

[32] Guo, L., Hu, X., Liu, W., & Liu, Y. (2025). Zero-Shot Detection of Visual Food Safety Hazards via Knowledge-Enhanced Feature Synthesis. Applied Sciences, 15(11), 6338.

[33] Wu, B., Qiu, S., & Liu, W. (2025). Addressing Sensor Data Heterogeneity and Sample Imbalance: A Transformer-Based Approach for Battery Degradation Prediction in Electric Vehicles. Sensors, 25(11), 3564.