International Journal of Innovative Computer Science and IT Research

Volume 01, Issue 03 – May-Jun 2025 • E-ISSN: 3067-1108 | P-ISSN: 3067-1094

Reinforcement Learning for Autonomous Decision-Making in Dynamic Computing Environments

Dr. Hector Geffner

Professor, RWTH Aachen University, Germany

Abstract

Modern computing environments are increasingly characterized by dynamic workloads, heterogeneous resources, fluctuating network conditions, distributed infrastructures, and continuously changing application requirements. Conventional resource management and decision-making mechanisms generally depend on predefined rules, static optimization models, or manually configured policies, limiting their ability to respond effectively to unpredictable operational conditions. Reinforcement Learning (RL) provides an adaptive computational paradigm in which intelligent agents learn decision policies through repeated interaction with an environment and feedback from observed rewards.

This paper examines the application of reinforcement learning for autonomous decision-making in dynamic computing environments, with particular emphasis on cloud computing, edge computing, distributed systems, resource scheduling, workload management, task offloading, and infrastructure optimization. The study reviews fundamental RL approaches, including Q-learning, Deep Q-Networks (DQN), policy-gradient methods, actor-critic architectures, Proximal Policy Optimization (PPO), and multi-agent reinforcement learning. A conceptual framework is developed in which environmental observations are transformed into states, candidate actions are evaluated through reward functions, and continuously updated policies determine autonomous computing decisions.

The analysis compares representative RL approaches according to adaptability, scalability, computational requirements, decision-making capability, and suitability for dynamic environments. Recent research demonstrates increasing use of deep and multi-agent reinforcement learning for edge-cloud scheduling and resource management, particularly where workload and network conditions vary rapidly. The paper further identifies challenges related to training cost, sample efficiency, safety, interpretability, partial observability, reward design, generalization, and deployment reliability. The findings indicate that RL can provide a strong foundation for self-adaptive computing systems when combined with efficient state representation, robust reward mechanisms, distributed learning, and appropriate safety controls. Future research should focus on explainable, federated, hierarchical, risk-aware, and resource-efficient reinforcement learning architectures capable of operating reliably across heterogeneous computing infrastructures.

Keywords: Reinforcement Learning, Autonomous Decision-Making, Deep Reinforcement Learning, Dynamic Computing, Cloud Computing, Edge Computing, Resource Management, Intelligent Systems, Task Scheduling, Multi-Agent Learning.

International Journal of Innovative Computer Science and IT Research

Volume 01, Issue 03 – May-Jun 2025 ● E-ISSN: 3067-1108 | P-ISSN: 3067-1094

1. Introduction

The rapid evolution of computing infrastructure has transformed conventional computing systems into highly dynamic and distributed environments. Cloud platforms, edge infrastructures, Internet of Things networks, mobile computing systems, and intelligent applications continuously experience variations in workload intensity, resource availability, network conditions, and service requirements. Such environments require computing systems to make timely decisions concerning resource allocation, task scheduling, workload placement, service scaling, and energy management. Traditional approaches frequently depend on predefined thresholds and manually engineered rules, which may perform adequately under predictable conditions but can become inefficient when environmental characteristics change rapidly.

Reinforcement Learning (RL) offers an alternative paradigm for autonomous decision-making because it enables computational agents to learn appropriate actions through interaction with an environment. Instead of relying entirely on predetermined instructions, an RL agent observes system conditions, selects an action, receives feedback through a reward signal, and updates its policy based on the observed outcome. This sequential learning process makes RL particularly relevant to computing environments in which decisions influence future system states. The approach can therefore support continuous adaptation rather than one-time optimization.

The increasing complexity of distributed computing has further encouraged the development of Deep Reinforcement Learning (DRL), which combines reinforcement learning with deep neural networks. DRL enables agents to process high-dimensional system states and learn nonlinear relationships between environmental conditions and suitable actions. Algorithms such as DQN, PPO, DDPG, SAC, and actor-critic models provide different mechanisms for learning decision policies. Recent research has applied these methods to edge-cloud scheduling, task offloading, resource allocation, and autonomous infrastructure management.

Dynamic computing environments introduce additional challenges because workloads may arrive unpredictably, resources may become temporarily unavailable, and network conditions can change independently of application behavior. In edge computing, for example, heterogeneous devices operate with different processing capacities and energy constraints. In cloud systems, large numbers of virtual machines and containers compete for computational resources. RL-based decision-making can continuously observe these variations and modify its policy accordingly. Recent work on edge-cloud environments demonstrates growing interest in multi-objective and multi-agent DRL approaches for simultaneously addressing latency, energy consumption, resource utilization, and service-level requirements.

Another important characteristic of autonomous computing is the need to balance short-term performance with long-term system objectives. A decision that minimizes immediate latency may increase energy consumption or resource contention later. Reinforcement learning naturally represents this trade-off through cumulative reward optimization. Consequently, an appropriately designed RL agent can learn policies that consider both current system performance and future consequences. However, this capability also introduces significant challenges involving reward engineering, training stability, exploration risks, and computational overhead.

This study therefore investigates reinforcement learning as a foundation for autonomous decision-making in dynamic computing environments. It examines theoretical foundations, application

Volume 01, Issue 03 – May-Jun 2025 • E-ISSN: 3067-1108 | P-ISSN: 3067-1094

areas, algorithmic approaches, implementation considerations, and major research challenges. The study also develops a conceptual analytical framework for understanding how RL agents can continuously observe computing environments, select actions, evaluate outcomes, and improve future decisions. The objective is to establish a comprehensive understanding of how reinforcement learning can contribute to adaptive, scalable, and intelligent computing infrastructures.

2. Objectives

2.1 Primary Objective

The primary objective of this study is to examine the role of reinforcement learning in enabling autonomous decision-making across dynamic computing environments. The study focuses on how RL-based agents can continuously learn from changing system conditions and optimize resource allocation, scheduling, workload placement, task execution, and infrastructure management while balancing performance, scalability, energy consumption, and service quality.

2.2 Evaluation of Reinforcement Learning Approaches

The first objective is to examine major reinforcement learning approaches used for autonomous computing decisions. Particular attention is given to Q-learning, DQN, policy-gradient methods, actor-critic architectures, PPO, and multi-agent reinforcement learning. Their characteristics are considered in relation to state representation, action selection, training complexity, scalability, adaptability, and suitability for heterogeneous computing environments.

2.3 Analysis of Dynamic Computing Applications

The second objective is to analyze how reinforcement learning can be applied to practical computing problems involving cloud resource management, edge computing, workload scheduling, task offloading, service scaling, energy optimization, and distributed infrastructure coordination. The analysis emphasizes the ability of RL systems to respond to unpredictable workload and infrastructure changes while maintaining acceptable performance.

2.4 Identification of Research Challenges

The third objective is to identify the major technical and operational challenges associated with deploying RL-based autonomous decision-making systems. These include sample inefficiency, training overhead, reward design, partial observability, safety, explainability, scalability, policy transfer, data requirements, and generalization across heterogeneous computing environments.

3. Review of Literature

3.1 Reinforcement Learning for Intelligent Decision-Making

Reinforcement learning has traditionally been formulated as a sequential decision-making problem in which an agent interacts with an environment by observing states, selecting actions, and receiving rewards. Q-learning established an important foundation for learning action-value functions without requiring a complete environmental model. With the development of deep neural networks, DQN extended this principle to high-dimensional state spaces. Subsequent actor-critic and policy-optimization methods improved the ability of RL agents to handle complex and continuous decision problems.

Volume 01, Issue 03 – May-Jun 2025 • E-ISSN: 3067-1108 | P-ISSN: 3067-1094

The suitability of RL for dynamic computing derives from its ability to optimize long-term outcomes rather than isolated decisions. Computing workloads often evolve continuously, meaning that resource allocation or scheduling decisions can affect subsequent system conditions. An RL agent can therefore learn policies that account for sequential consequences. However, efficient learning requires appropriate state representation, meaningful reward functions, sufficient training experience, and stable optimization procedures.

3.2 Deep Reinforcement Learning in Cloud and Edge Computing

Cloud and edge computing have become major application domains for DRL-based decision-making. Resource management problems frequently involve multiple variables, including CPU utilization, memory consumption, network bandwidth, workload arrival rates, task deadlines, energy requirements, and service-level objectives. DRL can integrate these variables into a learned policy and continuously adapt decisions according to observed conditions.

Recent studies indicate that DRL is increasingly being used for edge-cloud scheduling and resource optimization because these environments exhibit heterogeneous resources and highly variable workloads. Research has addressed task scheduling, resource allocation, task offloading, and SLA-aware optimization. A 2025 survey also identified provisioning, placement, scheduling, allocation, offloading, and load balancing as important RL-based resource-management applications in fog environments.

3.3 Multi-Agent Reinforcement Learning

Large-scale distributed computing environments cannot always be effectively controlled by a single centralized agent. Multi-Agent Reinforcement Learning (MARL) addresses this limitation by allowing multiple agents to learn simultaneously. Agents can represent edge nodes, cloud servers, applications, network components, or autonomous services.

MARL can support decentralized decision-making while enabling coordination among multiple computing entities. Recent research has explored multi-agent DRL for dynamic edge-cloud scheduling, including cooperative resource allocation and distributed task management. Nevertheless, MARL introduces additional challenges, including non-stationary learning environments, communication overhead, coordination complexity, and credit assignment.

4. Methodology

4.1 Research Design

This study adopts a conceptual and analytical research design to examine reinforcement learning as a mechanism for autonomous decision-making in dynamic computing environments. The methodology integrates theoretical analysis of reinforcement learning with a structured examination of its applications in cloud, edge, distributed, and heterogeneous computing systems. The study considers the relationship between environmental states, agent actions, reward signals, learning policies, and subsequent system conditions.

The analytical framework treats a computing infrastructure as a sequential decision environment. At each decision interval, an RL agent observes relevant system information and selects an action according to its current policy. The environment then changes according to workload, network, resource, and application conditions. The agent receives a reward reflecting performance objectives and updates

its policy using accumulated experience. This cycle enables continuous adaptation and autonomous optimization.

4.2 Reinforcement Learning Decision Model

The computing problem can be represented as a Markov Decision Process (MDP), defined by the tuple:

$M = (S, A, P, R, \gamma)$

where S represents the set of system states, A represents available actions, P represents transition probabilities, R represents the reward function, and γ represents the discount factor. The state may include CPU utilization, memory availability, workload intensity, network latency, energy consumption, queue length, and task priority. Actions may include resource allocation, task migration, scheduling, scaling, or task offloading.

The decision process can be represented through the following key components:

- **State observation:** Collect current computing and workload conditions.
- **Action selection:** Select an infrastructure or workload-management action.
- **Reward evaluation:** Measure the performance consequences of the selected action.
- **Policy update:** Improve the decision policy using observed experience.
- **Environment transition:** Move the computing system into a new state.
- **Continuous learning:** Repeat the process to improve future decisions.

The objective is to maximize cumulative discounted reward:

$G_t = \Sigma \gamma^k R_{t+k}$

where the cumulative reward represents the long-term value of decisions rather than only their immediate effects.

4.3 Algorithmic Framework

The proposed analytical framework considers different RL algorithms according to the nature of computing decisions. DQN is suitable for discrete actions such as selecting among predefined servers. Policy-gradient and actor-critic methods can address more complex or continuous actions. PPO provides a stable policy optimization mechanism, while SAC is particularly useful when exploration and continuous control are important. Multi-agent approaches become relevant when decisions are distributed among multiple computing nodes.

The framework follows these methodological stages:

1. Environmental monitoring
2. State construction
3. Action-space definition
4. Reward-function formulation
5. RL-agent training
6. Policy evaluation
7. Online adaptation
8. Safety and performance monitoring

Recent edge-cloud research has similarly explored PPO-based adaptive resource management and multi-objective scheduling to handle changing workload and infrastructure conditions.

International Journal of Innovative Computer Science and IT Research

Volume 01, Issue 03 – May-Jun 2025 • E-ISSN: 3067-1108 | P-ISSN: 3067-1094

4.4 Performance Evaluation Criteria

The effectiveness of autonomous RL decision-making should be evaluated using multiple performance indicators rather than a single optimization metric. Relevant measures include response latency, resource utilization, energy consumption, throughput, task completion rate, SLA compliance, operational cost, convergence speed, and policy stability. Multi-objective evaluation is especially important because optimizing one metric can negatively affect another.

Evaluation should also consider generalization to workload conditions that were not represented extensively during training. A reliable autonomous computing agent should maintain acceptable performance when workload intensity, network conditions, resource availability, or application requirements change. Recent studies emphasize scalability, robustness, fairness, SLA compliance, and energy efficiency as important considerations in dynamic edge-cloud RL systems.

5. Analysis

5.1 Reinforcement Learning Algorithms for Dynamic Computing

Different reinforcement learning algorithms provide different capabilities for autonomous computing decisions. Value-based algorithms such as DQN are effective when action spaces are discrete, while policy-gradient and actor-critic approaches provide greater flexibility for continuous decision problems. PPO offers stable policy updates, whereas multi-agent approaches support distributed decision-making.

Table 1: Comparative Characteristics of Reinforcement Learning Approaches for Dynamic Computing

RL Approach	Action Type	Major Strength	Major Limitation	Computing Application
Q-Learning	Discrete	Simple and interpretable	Poor scalability for large state spaces	Basic resource selection
DQN	Discrete	Handles high-dimensional states	Large training requirements	Server selection and scheduling
Policy Gradient	Continuous/Discrete	Direct policy optimization	Training variance	Resource allocation
Actor-Critic	Continuous/Discrete	Combines value and policy learning	Greater architectural complexity	Dynamic orchestration
PPO	Continuous/Discrete	Stable policy optimization	Computationally intensive	Autoscaling and scheduling

International Journal of Innovative Computer Science and IT Research

Volume 01, Issue 03 – May-Jun 2025 • E-ISSN: 3067-1108 | P-ISSN: 3067-1094

RL Approach	Action Type	Major Strength	Major Limitation	Computing Application
SAC	Continuous	Strong exploration capability	Complex implementation	Continuous resource control
Multi-Agent RL	Mixed	Distributed decision-making	Coordination complexity	Edge-cloud management

Interpretation: Table 1 demonstrates that no single reinforcement learning algorithm is universally optimal for dynamic computing. Algorithm selection should depend on the action space, system scale, degree of distribution, training resources, and required level of adaptability. DQN is appropriate for discrete decisions, whereas PPO and actor-critic methods provide greater flexibility for continuous resource-management problems. Multi-agent reinforcement learning is particularly appropriate when computing decisions must be distributed across multiple infrastructure nodes.

5.2 Autonomous Decision-Making in Dynamic Environments

Autonomous computing systems must respond to multiple environmental changes simultaneously. Workload spikes can increase queue lengths and processing delays, network fluctuations can alter task-offloading decisions, and resource failures can require rapid workload migration. RL enables decision policies to adapt based on these observed conditions rather than relying exclusively on fixed thresholds. Recent research illustrates this direction.

EdgeAutoOffloadAI combines PPO and a Transformer-based representation for dynamic task offloading and resource allocation in mobile edge computing, while incorporating fairness and tail-risk considerations. Other recent research has explored SLA-aware DRL scheduling and multi-objective scheduling that jointly considers latency, energy, and service-level requirements.

Table 2: Major Applications of RL-Based Autonomous Decision-Making

Application Area	Decision Problem	Relevant RL Capability	Expected Benefit
Cloud Computing	VM allocation and autoscaling	Adaptive policy learning	Improved utilization
Edge Computing	Task offloading	Real-time adaptation	Lower latency
Distributed Systems	Workload placement	Sequential optimization	Better load balancing
IoT Computing	Resource scheduling	Context-aware decisions	Energy efficiency
Container	Service scaling	Continuous control	Improved QoS

Volume 01, Issue 03 – May-Jun 2025 • E-ISSN: 3067-1108 | P-ISSN: 3067-1094

Application Area	Decision Problem	Relevant RL Capability	Expected Benefit
Platforms			
Network Computing	Routing and resource allocation	Dynamic policy selection	Network efficiency
Fog Computing	Task placement	Distributed learning	Reduced processing delay
Edge-Cloud Continuum	Multi-layer orchestration	Hierarchical decision-making	Better resource coordination

Interpretation: Table 2 shows that RL can support autonomous decision-making across multiple computing layers. Its major advantage is the ability to respond to changing system conditions while considering long-term consequences. The strongest applications are those in which workload behavior is difficult to predict and conventional static policies require frequent manual adjustment.

5.3 Autonomous Computing Architecture

An RL-based autonomous computing architecture can be organized into five interconnected layers. The monitoring layer collects telemetry from computing infrastructure. The state-processing layer converts raw measurements into meaningful representations. The learning layer contains the RL agent and policy model. The execution layer applies decisions to the infrastructure, while the feedback layer evaluates the consequences and returns reward information to the learning system.

Such an architecture can be centralized, decentralized, or hierarchical. Centralized architectures simplify coordination but may create scalability bottlenecks. Decentralized architectures improve autonomy and resilience but require coordination mechanisms. Hierarchical architectures can divide decisions between local and global agents. Recent research into hierarchical RL for computing-continuum management demonstrates the relevance of combining local agents with higher-level coordination.

6. Discussion

6.1 Adaptability and Intelligent Automation

The principal contribution of reinforcement learning to dynamic computing is its ability to support adaptive decision-making. Traditional rule-based systems require administrators to specify thresholds and responses in advance. In contrast, RL agents can learn policies from system interactions and continuously update their behavior. This characteristic is particularly useful when workload distributions and infrastructure conditions change over time.

Autonomous adaptation can reduce the need for frequent manual intervention. For example, an RL controller may learn to increase computing capacity during workload peaks and reduce resource allocation during low-demand periods. Similar principles can be applied to task scheduling, workload

Volume 01, Issue 03 – May-Jun 2025 • E-ISSN: 3067-1108 | P-ISSN: 3067-1094

migration, network routing, and energy management. Recent edge-cloud studies demonstrate that this adaptive capability is increasingly being investigated for real-world dynamic infrastructures.

6.2 Scalability and Distributed Decision-Making

Scalability remains one of the most important considerations in autonomous computing. As the number of devices, services, and computing nodes increases, centralized RL agents may face large state and action spaces. Multi-agent and hierarchical reinforcement learning provide potential solutions by distributing decision-making across multiple agents or levels.

However, decentralization introduces coordination challenges. Agents must learn how their individual decisions affect other agents and overall system performance. Communication between agents can also increase network overhead. Effective distributed RL systems therefore require efficient information sharing, parameter coordination, and reward mechanisms that balance local objectives with global system requirements.

6.3 Reliability and Practical Deployment

Although reinforcement learning offers strong adaptive capabilities, autonomous computing systems cannot rely solely on unconstrained exploration. An incorrectly selected action can cause service degradation, excessive resource consumption, or SLA violations. Consequently, practical deployment requires safety mechanisms that restrict harmful actions and monitor policy behavior.

Recent work increasingly incorporates fairness, SLA requirements, risk-aware optimization, and distributed learning into RL-based computing systems. These developments suggest that future autonomous computing architectures should treat RL as one component of a larger control framework rather than as an isolated optimization algorithm.

7. Challenges

The first major challenge is the computational cost associated with RL training. Deep reinforcement learning may require extensive interaction data and repeated policy updates before achieving stable performance. Training can become particularly expensive when the computing environment contains numerous states, actions, and interacting entities. Simulation environments can reduce deployment risk, but discrepancies between simulated and real infrastructures may reduce policy effectiveness. Efficient offline training, transfer learning, experience reuse, and model compression are therefore important for practical deployment.

A second challenge concerns reward-function design. Computing environments frequently have multiple competing objectives, including latency, energy consumption, resource utilization, cost, fairness, and reliability. A reward function that overemphasizes one metric can produce undesirable system behavior. Designing balanced multi-objective rewards is consequently a central research problem. Recent studies increasingly incorporate SLA, fairness, latency, and energy considerations into RL-based scheduling frameworks.

A third challenge is reliability under unseen environmental conditions. RL policies can perform well within training distributions but degrade when workload patterns, resource configurations, or network conditions change substantially. Dynamic computing environments are especially vulnerable to such distribution shifts. Robust learning, continual adaptation, uncertainty estimation, safe exploration,

Volume 01, Issue 03 – May-Jun 2025 • E-ISSN: 3067-1108 | P-ISSN: 3067-1094

explainability, and policy verification are therefore necessary for dependable autonomous computing systems.

8. Future Research

8.1 Explainable and Trustworthy Reinforcement Learning

Future research should develop explainable RL techniques capable of identifying why particular infrastructure decisions are selected. Explainability is important for administrators who need to validate resource-management actions and diagnose unexpected behavior. Combining interpretable policy representations with deep learning may provide a balance between decision-making capability and operational transparency. Research should also investigate uncertainty-aware RL, policy verification, and safety constraints to ensure that autonomous decisions remain within acceptable operational boundaries.

8.2 Federated and Privacy-Aware Reinforcement Learning

Distributed computing environments increasingly operate across organizations and administrative domains. Centralizing operational data for RL training may create privacy, security, and data-governance concerns. Federated reinforcement learning provides an alternative in which agents can learn locally while sharing selected model information rather than raw operational data. Recent research has already explored federated RL for multi-tenant edge resource allocation. Future work should investigate communication-efficient aggregation, heterogeneous client participation, privacy protection, and robustness against malicious model updates.

8.3 Hierarchical and Multi-Agent Autonomous Computing

Future autonomous computing infrastructures are likely to combine hierarchical and multi-agent learning. Local agents could manage individual servers or edge nodes, while higher-level agents could coordinate global objectives such as energy efficiency, workload distribution, and SLA compliance. Hierarchical reinforcement learning can reduce decision complexity by separating strategic and operational decisions. Research should investigate scalable coordination mechanisms, adaptive communication policies, decentralized training, and cross-layer optimization for large computing continuums.

9. Conclusion

Reinforcement learning represents a powerful approach for developing autonomous decision-making capabilities in dynamic computing environments. Unlike static rules and conventional optimization procedures, RL enables computational agents to learn from repeated interaction with changing environments and improve decisions according to long-term objectives.

This capability is highly relevant to cloud computing, edge computing, distributed systems, IoT infrastructures, and computing-continuum architectures in which workload patterns, resource availability, network conditions, and service requirements change continuously. DQN, policy-gradient, actor-critic, PPO, SAC, and multi-agent reinforcement learning each provide distinct capabilities for addressing different decision-making requirements. The analysis demonstrates that algorithm selection should depend on the characteristics of the computing environment and the nature of the decision

problem. The study further demonstrates that RL can support autonomous resource allocation, task scheduling, workload placement, task offloading, service scaling, energy management, and infrastructure coordination. Recent research confirms growing application of DRL to dynamic edge-cloud environments, including SLA-aware scheduling, multi-objective optimization, adaptive offloading, and distributed resource management.

Nevertheless, practical deployment requires careful consideration of training costs, reward design, scalability, partial observability, safety, explainability, and generalization. Autonomous decision-making should therefore be implemented within controlled architectures that combine learning capabilities with monitoring, policy constraints, and operational safeguards.

Future autonomous computing systems are expected to move toward more distributed, hierarchical, privacy-aware, and risk-sensitive learning architectures. Federated reinforcement learning can help address data-governance requirements, while multi-agent and hierarchical methods can improve scalability across distributed infrastructures. Explainable and safe RL can increase administrator confidence and reduce operational risks.

References

1. Aghapour, Z., Sharifian, S., & Taheri, H. (2023). Task offloading and resource allocation algorithm based on deep reinforcement learning for distributed AI execution tasks in IoT edge computing environments. *Computer Networks*, *223*, 109577. https://doi.org/10.1016/j.comnet.2023.109577
2. Arulkumaran, K., Deisenroth, M. P., Brundage, M., & Bharath, A. A. (2017). Deep reinforcement learning: A brief survey. *IEEE Signal Processing Magazine*, *34*(6), 26–38. https://doi.org/10.1109/MSP.2017.2743240
3. Bellemare, M. G., Dabney, W., & Munos, R. (2017). A distributional perspective on reinforcement learning. *Proceedings of the 34th International Conference on Machine Learning, 70*, 449–458.
4. Dulac-Arnold, G., Levine, N., Mankowitz, D. J., Li, J., Paduraru, C., Gowal, S., & Hester, T. (2021). Challenges of real-world reinforcement learning: Definitions, benchmarks and analysis. *Machine Learning, 110*, 2419–2468. https://doi.org/10.1007/s10994-021-05961-4
5. Haarnoja, T., Zhou, A., Abbeel, P., & Levine, S. (2018). Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. *Proceedings of the 35th International Conference on Machine Learning, 80*, 1861–1870.
6. Hessel, M., Modayil, J., van Hasselt, H., Schaul, T., Ostrovski, G., Silver, D., & Hasselt, H. (2018). Rainbow: Combining improvements in deep reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence, 32*(1), 3215–3222. https://doi.org/10.1609/aaai.v32i1.11796
7. Li, X. (2022). 5G converged network resource allocation strategy based on reinforcement learning in edge cloud computing environment. *Computational Intelligence and Neuroscience, 2022*, 6174708. https://doi.org/10.1155/2022/6174708
8. Lillicrap, T. P., Hunt, J. J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D. (2016). Continuous control with deep reinforcement learning. *International Conference on Learning Representations*.
9. Luo, Z., & Dai, X. (2024). Reinforcement learning-based computation offloading in edge computing: Principles, methods, challenges. *Alexandria Engineering Journal*.

10. Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., & Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature, 518*, 529–533. https://doi.org/10.1038/nature14236
11. Penney, D., Li, B., Sydir, J., Chen, L., Tai, C., Lee, S., Walsh, E., & Long, T. (2023). PROMPT: Learning dynamic resource allocation policies for network applications. *Future Generation Computer Systems, 145*, 164–175. https://doi.org/10.1016/j.future.2023.03.016
12. Poltronieri, F., Stefanelli, C., Suri, N., & Tortonesi, M. (2022). Value is King: The MECForge deep reinforcement learning solution for resource management in 5G and beyond. *Journal of Network and Systems Management, 30*, 63. https://doi.org/10.1007/s10922-022-09672-6
13. Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv*. https://arxiv.org/abs/1707.06347
14. Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., & Hassabis, D. (2016). Mastering the game of Go with deep neural networks and tree search. *Nature, 529*, 484–489. https://doi.org/10.1038/nature16961
15. Wang, L., Wu, J., Gao, Y., & Zhang, J. (2023). Deep reinforcement learning based resource allocation for cloud native wireless network. *arXiv*. https://arxiv.org/abs/2305.06249
16. Wu, J., Huang, C., Huang, H., Lv, C., Wang, Y., & Wang, F.-Y. (2024). Recent advances in reinforcement learning-based autonomous driving behavior planning: A survey. *Transportation Research Part C: Emerging Technologies, 164*, 104654. https://doi.org/10.1016/j.trc.2024.104654
17. Zhou, G., Tian, W., Buyya, R., Xue, R., & Song, L. (2024). Deep reinforcement learning-based methods for resource scheduling in cloud computing: A review and future directions. *Artificial Intelligence Review, 57*, 124. https://doi.org/10.1007/s10462-024-10756-9
18. Zhou, Z., Chen, X., Li, E., Zeng, L., Luo, K., & Zhang, J. (2019). Edge intelligence: Paving the last mile of artificial intelligence with edge computing. *Proceedings of the IEEE, 107*(8), 1738–1762. https://doi.org/10.1109/JPROC.2019.2918951
19. Mampage, A., Karunasekera, S., & 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
20. Yin, Y., Cui, L., & others. (2023). An reinforcement learning approach for allocating software resources. *Concurrency and Computation: Practice and Experience*. https://doi.org/10.1002/cpe.6349