Journal of Electronics and Informatics (ISSN: 2582-3825)
www.irojournals.com/iroei/

Analysis of Learning-Based Methods for Intelligent Edge Computing

Nay Myo Sandar[1*], Thinzar Aung Win[2]

Lecturer, Faculty of Business and Technology, Stamford International University, Thailand.

E-mail: [1*]naymyo.sandar@stamford.edu, [2]thinzaraung.win@stamford.edu

Abstract

Edge Computing is recognized as an important computing paradigm that helps in the processing of data as well as the making of decisions near the user, hence reducing the need to rely on cloud infrastructure. Edge Computing is of high importance in fields that are sensitive to latency, such as smart vehicles, IoT services, health monitoring, UAVs, and smart cities. Edge Computing environments face various issues, such as limited computational power, network variability, mobility, and the need to process high volumes of diverse tasks. These issues in edge computing are difficult to handle using traditional methods of optimization. Deep Reinforcement Learning (DRL) algorithms are significantly used in edge computing systems as an intelligent decision-making tool that can optimize decisions through interactions with the edge computing system. This review paper presents a comprehensive analysis of recent studies and representative research that apply DRL techniques to edge computing systems. The reviewed papers cover various applications such as vehicular edge computing, IoT-based multi-access edge computing, UAV-assisted edge computing, collaboration between fog and cloud computing, and blockchain and federated learning-based edge computing systems. The discussion on DRL applications encompasses task offloading, resource management, workload management, caching, and mobility management. A comparative analysis discusses the most used DRL algorithms and their applications along with performance improvements. Additionally, this literature review highlights some of the main drawbacks of DRL, such as training complexity, scalability, privacy, and deployment, and identifies areas that require improvement to create safe and efficient DRL-based edge computing solutions.

*** Corresponding Author**

DOI: https://doi.org/10.36548/jei.2025.4.003
Received: 12.11.2025, received in revised form: 10.12.2025, accepted: 27.12.2025, published: 07.01.2026

Keywords: Edge Computing; Deep Reinforcement Learning, Task Offloading, Resource Allocation, Internet of Things (IoT), Vehicular Edge Computing.

1. Introduction

Edge Computing is an emerging technology where processing occurs close to the end user, without the need for transferring the entire data to cloud computing resources located in remote clouds. This offers advantages in terms of latency, bandwidth efficiency, and the capability to improve response speed for applications such as autonomous cars, medical services, industrial operations, home automation, and smart cities. With the growth of connected devices, particularly after the rise of the Internet of Things (IoT), there has been an accelerated demand for speed accompanied by smarter data processing near the edges. This demand has been met through the intention of Edge Computing, aiming to provide computational strength closer to the end user and reduce reliance on the clouds.

There are several challenges associated with Edge Computing. Firstly, devices typically have lower processing power compared to servers in the cloud. Additionally, changes in signal strength, interference, and mobility in wireless networks have been observed, even in cases involving vehicles or drones. Edge Computing must handle various tasks from users, which often need attention at odd hours. These erratic and turbulent environments complicate the application of conventional optimization techniques, which generally rely on optimization-based approaches and heuristics that may not effectively respond to changes in such environments. To address the challenges associated with Edge Computing, the technical community has turned to Deep Reinforcement Learning, a methodology that leverages neural networks and relies on reinforcement learning for well-informed decision-making through experience. Furthermore, Deep Reinforcement Learning can learn optimal task offloading, resource scheduling, cache management, and mobility strategies without needing to model the environment in advance. In this paper, the term Deep Reinforcement Learning (DRL) is consistently used to represent deep learning-assisted reinforcement learning methods, and any alternative terminology has been removed for clarity. These methods include greedy scheduling, Lyapunov optimization, and fixed rule-based decision models that rely on predefined system assumptions and static network conditions.

This allows it to be well adapted to very dynamic edge computing applications. Most recent reports have indicated that DRL demonstrates significant performance improvement in minimizing delay, enhancing energy efficiency, boosting the success rate of tasks, and improving the overall Quality of Service (QoS) to edge users. With the current pace of research development in this field, this review article evaluates 25 notable studies that apply DRL to edge computing problems. The analysis of these works also discusses the strengths, weaknesses, and future prospects of the literature to expand the development of larger, efficient, scalable edge computing architectures based on the DRL approach for better edge networks. Traditional optimization or heuristic approaches, including scheduling schemes, Lyapunov optimization, and rule-based offloading strategies, have shown visible performance degradation in the edge environment. It clearly states that various existing literature results confirm that traditional optimization strategies or rule-based strategies experience noticeably large end-to-end delays and energy consumption costs during increased user mobility and task arrivals, compared to approaches based on learning strategies [1], [3], [5], [9], [22].

The implementation of 5G networks and newer technologies of 6G makes intelligent edge computing systems even more valuable. These networks have very low latency and high reliability needs that cannot be fulfilled by cloud computing alone. DRL provides the capability to evolve continuously and learn swiftly in the network environment, which is why it is appropriate for the next-generation communication environment. Additionally, different edge devices may become more heterogeneous, and DRL can facilitate tailored decision-making regarding the capabilities of these devices. The possible use of DRL in handling complex edge conditions that involve multi-agent interactions among vehicles, sensors, and UAVs is also highlighted in many existing solutions. Federated learning combined with DRL has also proven to improve privacy by minimizing communication overhead. Such developments indicate that edge computing based on DRL will improve in the next few years. As such, a systematic review needs to be undertaken to identify the latest trends and determine the areas where further research is still required to assess performance improvements.

This helps ensure that future models for DRL will be much more operationally oriented and capable of dealing with the practical needs of edge computing. There are different algorithms for DRL depending on the nature of the problem. DQN is suitable for discrete offloading decisions, while DDPG and actor-critic methods handle continuous resource control. PPO is preferred for stability under dynamic environments.

2. Literature Survey

Edge Computing for Smart Vehicles has been majorly examined by researchers. Li et al. [1] proposed a Deep Deterministic Policy Gradient (DDPG)-based collaborative task offloading system for vehicular networks, where tasks are divided and processed by nearby edge servers to reduce service delay and task failures. Liu et al. [5] also focused on vehicle edge computing and designed a reinforcement learning-based solution to offload tasks to nearby vehicles to improve service quality under dynamic road conditions.

Similarly, Ning et al. [6] proposed an intelligent offloading system that takes into consideration communication and computing resources to enhance user experience in fast-changing vehicular environments. Qiao et al. [10] optimized cooperative caching in vehicular edge networks using DRL, which successfully reduces access delays and improves content delivery performance. Recently, Geng et al. [24] developed a distributed DRL-based offloading system to reduce latency and energy consumption in a vehicle-assisted edge computing network. These studies demonstrate that DRL can help vehicles make better offloading decisions more quickly.

In the IoT domain, Zhao et al. [2] used DRL to offload tasks from mobile devices to edge servers, which reduces power consumption and delays in smart IoT applications. Chen et al. [3] used DQN to control offloading in congested mobile networks, demonstrating that wireless link and computing resources can be effectively optimized using DRL. Lei et al. [23] proposed a deep RL-based system for joint resource management in IoT edge computing, aimed at improving delay and energy performance with random task arrivals.

These approaches are effective as long as the workload of numerous IoT devices is unpredictable and sent to edge servers. Workload scheduling has also been improved through reinforcement learning. Zheng et al. [9] proposed a DRL workload scheduling algorithm that distributes edge server workload and reduces the number of failed tasks. Sheng et al. [15] focused on IoT edge scheduling and showed that RL-based scheduling improves the number of tasks completed on time. These contributions support the fact that DRL assists edge computing in handling multiple tasks more efficiently.

Liu et al. [12] proposed DRL-based flight path planning for UAV edge computing to achieve stable communications when serving ground users. Wang et al. [16] optimized the movements of UAVs using an actor-critic method to reduce the energy consumption of mobile users and achieve better Quality of Service. DRL helps UAV-assisted edge computing systems remain efficient even when devices and UAVs are in motion. He et al. [7] used DRL with a blockchain for secure allocation of computing resources in IoT networks and maintaining the security of the data. Yu et al. [25] enhanced privacy even further by combining federated learning and DRL for multi-access edge computing, reducing reliance on centralized data and optimizing resources.

The reviewed papers were in agreement that DRL is effective in solving the main challenges in edge computing. This enhances decision-making regarding allocation, offloading, scheduling, and mobility-aware services. However, many works still face problems with high training complexity, generalization of models, and low generalization of models in the real world. Therefore, future studies must focus on making DRL faster, more secure, and easier to use in practical edge systems.

The existing studies are related to Vehicular Edge Computing (VEC) and IoT-based Multi-Access Edge Computing (MEC). Very few are aimed at UAV-based edge computing, Fog/Cloud collaborations, and secure DRL-enabled edge systems. This reveals opportunities for research in less-studied areas such as mobility-aware UAV coordination and enhanced edge learning. These studies are analyzed comparatively by examining their algorithmic choices, application scenarios, and reported performance outcomes. The discussion highlights common strengths, limitations, and emerging trends across vehicular, IoT, UAV-assisted, and secure edge computing environments. This analytical perspective enables the identification of research gaps related to scalability, generalization, and deployment feasibility rather than merely listing studies in isolation.

3. Research Gap

Although a high potential for Deep Reinforcement Learning (DRL) is found in improving various aspects of edge computing, there are some research gaps that have yet to be addressed in the studies of DRL. Most of the implemented literature on DRL consists of simulation models, which do not encompass various realistic aspects of this sector, including wireless communication variability, hardware failures, irregular movements of users, and energy variability. The practical implementation of DRL at the edge has yet to be fully executed, along with the generalizability of these simulation models. Another important gap is the lack of scalability of the proposed DRL approaches. Many existing works in the literature use single-agent or centralized DRL models, which are extremely inefficient or unstable when the number of edge devices increases. However, distributed learning is required in large-scale IoT, vehicular, and drone systems, though there is less work in the existing literature that investigates multi-agent DRL models. In addition, the existing DRL models consume high memory and involve long training times, which are not suitable for edge devices, as depicted in Fig. 1.

Blockchain and federated learning are sometimes used in the work, but these solutions are not common, and many DRL models rely on centralized training, where user data may be revealed.

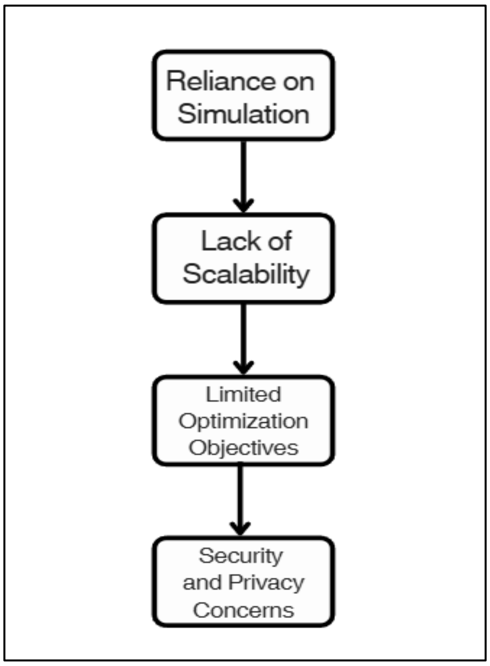

Figure 1. Research Gap in DRL-based Edge Computing

The value addition in this review is that it points out all these limitations and provides a collective review on vehicles, IoT, UAV, fog, and secure edge systems. By comparing the results and implementation of DRL techniques, this paper articulates in which ways the current work is performing well and what kind of important work might be required for success. The results clearly show that a practical, adaptive, and privacy-respectful DRL technique is required to deliver future edge computing technology. Figure 1 explains key limitations in the existing DRL edge computing work, which include high complexity during training and a lack of support for privacy. The figure above indicates that there is a need for distributed and privacy-friendly DRL with lower complexity for the future edge scenario. The limitations exposed by the review work may be ranked according to their effect on deployment. The most important limitation is that the central DRL model is neither scalable nor generalized in a large-scale edge computing scenario. The next limitation is high complexity during training and computational cost, which hinders execution on edge devices. Limitations in the areas of privacy, security, and the advantages of collecting information from the central node are another area of concern. Lastly, the absence of experimentation in edge computing is another challenge that, when prioritized, will be useful in the improvement of DRL in edge computing.

4. Comparative Analysis

From the comparative analysis illustrated in Table 1, it is well understood that different methods of Deep Reinforcement Learning are used for various edge computing-related problems in different environmental setups, such as vehicle networks, IoT networks, drone-assisted communication networks, and secure edge computing networks. This review reveals the effectiveness of DRL by considering the objective, methodology, and performance improvement of each existing research work in minimizing delay, reducing energy consumption, and managing network resources, as well as enhancing the completion of tasks in real time. The selection of DRL algorithms used for carrying out the comparative analysis is based on the relevance of the algorithms with respect to particular decision-making-related problems. For example, value-based algorithms such as Deep Q-Network (DQN)-based algorithms can be applied to decision-making problems involving discrete decisions, such as binary offloading; algorithms based on continuous control, such as Deep Deterministic Policy Gradient, can be applied to decision-making problems involving power allocations; algorithms that provide support for stable updates toward policy, such as Proximal Policy Optimization, can be applied to decision-making problems involving highly dynamic processes, such as resource allocations in edge computing.

The above table reveals that DRL algorithms such as DQN, DDPG, PPO, and actor-critic algorithms are most efficient, as they address mobility, unpredictable workloads, and computational constraints in an edge setting. It also highlights gaps such as the high cost of training, deployment challenges, and lack of real-world testing.

Therefore, this comparative analysis supports the purpose of this study in demonstrating the contribution of DRL techniques in solving edge computing challenges and assisting future research in designing smarter, faster, and more practical solutions for edge intelligence. It is important to note that the performance improvements reported in the reviewed studies are derived from simulation-based evaluations conducted under different assumptions, network models, and workload configurations. Consequently, there is no valid methodology for comparing studies numerically. This review thus focuses on performance comparisons and not absolute improvements in order to maintain academic reproducibility.

Table 1. Comparative Analysis Table

Authors	Edge Computing Scenario	DRL Technique Used	Objective of Study	Result / Improvement
Li et al [1]	Vehicular Edge Computing	DDPG	Collaborative task offloading	Reduced delay and failure rate
Zhao et al [2]	IoT Mobile Edge Computing	DQN	Smart offloading for IoT devices	Lower energy and delay
Chen et al [3]	Ultra-dense MEC	DQN	Dynamic offloading decisions	Improved wireless + compute efficiency
Tang & Wong [4]	Mobile Edge Computing	Double DQN	Delay-sensitive task decisions	Better decision accuracy
Liu et al [5]	Vehicular Edge Computing	Q-Learning + DRL	Resource allocation and offloading	Improved QoS with mobility
Ning et al [6]	Vehicular Edge Computing	DRL	QoE improvement via scheduling	Better user experience
He et al [7]	Blockchain-IoT Edge	A3C	Secure allocation of resources	Higher security and trust
Huang et al [8]	Wireless Powered MEC	DNN + RL	Offloading and power control	Reduced energy consumption
Zheng et al [9]	Edge Cloud Computing	DQN	Workload scheduling	Reduced failures and execution time
Qiao et al [10]	Vehicular Edge Caching	DDPG	Cooperative caching	Lower content delay
Li et al [11]	Distributed MEC	DDPG	Distributed offloading	Faster decision with low delay

Liu et al [12]	UAV-mounted MEC	DRL	Path planning and user support	Better connection stability
Zhan et al [13]	Vehicular Edge	PPO	Offloading scheduling	Balanced load + lower cost
Alfakih et al [14]	Mobile Edge	SARSA-DRL	Resource allocation	Reduced overhead
Sheng et al [15]	IoT Edge	RL (policy-gradient)	Task scheduling	More tasks completed on time
Wang et al [16]	UAV-assisted MEC	Actor-Critic RL	Trajectory control	Reduced user energy
Yamansavascilar et al [17]	General Edge	DDQN	Task orchestration	Better service latency
Luo et al [18]	Vehicular Edge Data	DQN	Cooperative data scheduling	Improved data delivery
Peng & Shen [19]	Vehicular MEC	DDPG (hierarchical)	Multi-resource management	Better QoS under road mobility
Goudarzi et al [20]	Fog/Edge Placement	Implicit DRL	Application placement	Improved delay for DAG services
Zhang et al [21]	UAV-assisted Edge	Actor-Critic	Drone-assisted offloading	Higher throughput and QoE
Chen & Wang [22]	Multi-user MEC	DDPG	Decentralized offloading	Lower power-delay cost
Lei et al [23]	IoT Edge (NB-IoT)	Value-based DRL	Joint resource control	More delay-efficient system
Geng et al [24]	Vehicular Edge	Multi-agent DRL	Distributed offloading	Lower latency & energy use
Yu et al [25]	Multi-access Edge + FL	Two-timescale DRL	Offloading + caching	Reduced execution time + privacy

5. Motivation of Further Research

The above literature has proven that the key benefit of DRL in an edge computing system is its advantage. Results from various research studies have highlighted that DRL can contribute to a reduction in waiting times for services, energy consumption, and failure incidents related to tasks, in contrast to traditional approaches, to varying extents. An issue that exists when DRL is implemented in edge computing is the sparsity in reward schemes. The PPO and DDPG approaches in the policy gradient technique are highly resistant to environments that exhibit sparsity in rewards since they adjust their policies based on their expectations rather than on rewards. This slows down convergence for algorithms like DQN.

Using this example, Geng et al. [24] showed that their distributed DRL-based offloading approach is efficient in reducing both latency and energy consumption for fine-grained tasks offloaded in a vehicular edge network. Using this approach, Li et al. [1] also demonstrated that the decision model and DDPG are capable of overcoming the difficulties introduced in vehicular systems due to mobility and dynamic network variations, resulting in improved user delay performance. Additionally, in discussing workload scheduling, the performance results of strategies employing DDPG, as proposed by Chen et al. [22], were also provided, indicating better performance compared to strategies employing DQN and Power-Delay trade-off. Zheng et al. achieved the best performance in workload scheduling, effectively minimizing service time and the failure percentage of tasks in scenarios with a large number of users.

A comparative evaluation of the distributed DRL scheduling scheme, X-DDRL, in fog/edge environments demonstrated that the technique is 30% faster in execution, 11% more efficient in power savings, and 24% lower in overall cost when compared to high-growth DRL benchmarks. These results convey that, contrary to traditional optimization techniques, DRL can adapt to variations in the network, mobility, and resource availability in real-time. Challenges remain, including long training times, size, and inconvenience in model transitions based on real-world situations. Overall, DRL enhances the intelligence level of edge computing and the optimization process, although further improvements are desired in the applications seen in Table 2. It should be noted that the performance advantages mentioned in the review are obtained from simulation-based analyses conducted in the concerned studies. These analyses consider synthetic workloads, mobility models, and pre-figures in the relevant environments. Hence, the results obtained should be assumed to convey the intended comparisons and nothing more.

Table 2. Performance Metrics Comparison Across Studies

Scenario	Benchmark Compared	Key Metric Improvements
Vehicular Edge [24]	Conventional RL	Lower latency and energy consumption

MEC Vehicular [1]	Non-learning approach	Better low-latency response under mobility
Multi-user MEC [22]	DQN & Greedy	Lower power cost with maintained delay
Edge Workload [9]	Traditional schedulers (Greedy scheduling, heuristic-based scheduling)	Lowest service time & failure rate
Fog/Edge DAG Placement [20]	PPO & Double-DQN	Reported faster execution and lower energy consumption under DAG-based workloads
MEC Vehicular [19]	Traditional optimization (Lyapunov optimization, rule-based allocation)	Higher QoS satisfaction ratios

This table provides an overview of how various DRL methods contribute to the enhancement ofEdge Computing in the application domain. The more advanced technique upon which this is based can be referred to as Vehicular Edge Computing, which is also impacted by enhanced movementand network state transitions, resulting in a decrease in delay values. This, in turn, leads to the observation of energy efficiency phenomena. In scenarios involving IoTor MEC Multi-user environments, delays caused by time interval execution are reduced, as is the application of low energy due to the nature of resource-limited devices. The entire process of course scheduling in Deep Learning Inception enhances Virtual Machines, which is one of the reasons for success in reducing task failure rates. For UAV-supported Edge Computing, the main goal ofDRL is energy efficiency and a robust communication link during movement. Among all the components available in the framework for performance metrics, those related to End-to-End Latency and Task Success Ratio are most relevant in terms of quality of service (QoS) usage value, and they can even be expanded to include latency-sensitive services in Vehicular Edge Networks and HealthcareEdge Networks. Energy consumption directly affects the sustainability process. Indeed, energyconsumption is a crucial factor in addressing usage issues for both IoT-based networks and UAV-enabled networks.

To facilitate collaboration between Fog and Cloud for quicker and more cost-efficient edge execution, an enhanced DRL approach is employed [19][22][24]. Researchers will find it easy to use this table to identify which DRL approaches can be adopted when and the effective paths that need to be taken in future research to achieve greater performance in the actual edge system, as presented in Table 3.

Table 3. Thrust Areas and Research Directions in DRL-based Edge Computing

Scenario	DRL Algorithms Commonly Used	Reported Performance Trends
Vehicular Edge Computing (VEC)	DDPG, DQN, Actor-Critic, Multi-agent DRL [1], [19], [24]	Significant latency reduction and improved energy efficiency under high mobility
IoT & Multi-user MEC	DQN, DDPG, SARSA [22], [23]	Improved energy efficiency and faster task execution under dynamic workloads
Workload Scheduling	DQN, PPO variants [9], [15]	Lower task failure rates and better resource utilization
UAV-assisted Edge Computing	Actor-Critic, DRL Path Planning [12], [16], [21]	Enhanced connectivity stability and energy-aware trajectory control
Fog/Cloud + Edge Collaboration	Double-DQN, PPO-based DRL [20]	Improved execution time and balanced computation cost
Secure & Privacy-aware Edge	A3C, Two-timescale DRL, Federated DRL [7], [25]	Enhanced privacy and security with stable latency performance

Table 3 provides a combined overview of the applications of DRL in various edge computing settings, rather than a unified common testbed [4] [7]. The performance improvement achievements are referenced from various independent research works that target a common set of scenarios, such as vehicular edge computing, IoT-enabled MEC systems, UAV-enabled edge computing, fog and edge collaboration, and secured edge computing systems. Although DRL consistently achieves valid delay improvements, energy conservation, and proper utilization of resources, the reviewed research works are compared based on various network settings, scenarios, and mobility patterns [11] [17]. The main gaps and deficiencies are uncovered from the different networks and settings, and based on that, Table 3 provides directions towards solutions like multi-agent DRL methods to manage mobility-intensive applications, lightweight DRL algorithms to be applied on various resource-deficient edge nodes, distributed learning techniques to overcome scalability issues, and federated learning for protecting privacy. Therefore, this table not only draws comparisons based on performance trends but also serves as an aid for future studies on deployable DRL-based edge computing architectures. Scalability and generalization gaps serve as motivations for employing multi-agent/DRL architectures. Large training complexity and resource scarcity motivate the leveraging of lightweight and edge-conscious DRL models. Privacy-related concerns act as incentives for incorporating Federated and Privacy-Preserving Learning techniques, while deployment difficulties underscore the need for laboratory testbeds and experimental assessments [20] [23]. This systematic research approach leads to each thrust area having an inherent vision that specifically seeks to overcome an identified shortcoming in existing research efforts. These postulations align with recent reports in existing DRL-based edge computing research efforts [1] [9] [20] [24]. This table draws qualitative comparisons on performance trends based on individual research efforts. There is no basis for comparing absolute performance improvements because simulation configurations differ.

Table 4. Identifies Research Gaps and Solutions

Thrust Area	Scenario	Identified Research Gap	Suggested Direction
DRL Offloading	Vehicular Edge	Poor generalization under mobility [1], [5], [24]	Mobility-aware multi-agent DRL
Resource Allocation	IoT & MEC	High training cost [3], [14], [23]	Lightweight DRL models
Scheduling	Edge Servers	Task failures at peak load [9], [15]	Priority-aware DRL
UAV Edge	UAV-assisted MEC	Scalability issues [12], [16], [21]	Distributed actor-critic
Secure Edge	Blockchain & FL	Privacy leakage [7], [25]	Federated DRL

This table.4 summarizes the key thrust areas identified from the reviewed literature and highlights the major research gaps across different edge computing scenarios. It clearly links each gap to a suitable future research direction, providing a structured roadmap for developing more scalable, adaptive and efficient DRL-based edge computing solutions.

Table 5. Advantages and Limitations of Deep Reinforcement Learning Techniques in Edge Computing

Aspect	Advantages	Limitations
DRL Adaptivity	Learns dynamic policies [3], [6], [24]	Slow convergence [3], [20]
Resource Control	Efficient utilization [1], [8], [22]	High training cost [14], [23]
Scalability	Handles complex states [11], [17]	Centralized DRL bottleneck [20], [24]
Deployment	Model-free learning [4], [18]	Mostly simulation-based [9], [17]

The above table.5 presents a concise comparison of the main advantages and limitations of DRL when applied to edge computing environments. The findings suggest that both the strengths of DRL, such as adaptive decision-making and efficient resource control and the challenges that must be addressed for practical deployment.

5.1 Limitations of this Review

This review is limited to peer-reviewed studies evaluated primarily through simulations. Differences in experimental setups, datasets and assumptions make direct numerical comparison challenging. Industrial deployments and real-world testbeds are outside the scope of this study.

6. Conclusion

The application of DRL algorithms in the optimization of the performance of edge computing systems has been comprehensively surveyed in the review paper. The role of the survey paper is to present the application survey of the DRL-based solutions in the core operational functionalities, such as task offloading, resource allocation, and scheduling in different use cases like Vehicular Edge Computing and IoT-Based Multi-Access Edge Computing systems. This survey highlights the benefits of DRL solutions over conventional solutions in the edge computing environment through their ability to improve levels of latency, energy efficiency, task completion time, and QoS. Limitations, such as variation in data distribution requirements, do not significantly impact the contributions, but they emphasize key challenges in detail, such as the complexity of testing and training in simulation-based testing and the challenges regarding scalability and privacy issues, respectively, in the deployment of DRL solutions in edge computing systems. A future study roadmap is provided within the paper.

References

[1] Li, Mushu, Jie Gao, Lian Zhao, and Xuemin Shen. "Deep Reinforcement Learning for Collaborative Edge Computing in Vehicular Networks." IEEE Transactions on Cognitive Communications and Networking 6, no. 4 (2020): 1122-1135.

[2] Zhao, Rui, Xinjie Wang, Junjuan Xia, and Liseng Fan. "Deep Reinforcement Learning Based Mobile Edge Computing for Intelligent Internet of Things." Physical Communication 43 (2020): 101184.

[3] Chen, Xianfu, Honggang Zhang, Celimuge Wu, Shiwen Mao, Yusheng Ji, and Mehdi Bennis. "Performance Optimization in Mobile-Edge Computing Via Deep Reinforcement Learning." In 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), IEEE, 2018, 1-6.

[4] Tang, Ming, and Vincent WS Wong. "Deep Reinforcement Learning for Task Offloading in Mobile Edge Computing Systems." IEEE Transactions on Mobile Computing 21, no. 6 (2020): 1985-1997.

[5] Liu, Yi, Huimin Yu, Shengli Xie, and Yan Zhang. "Deep Reinforcement Learning For Offloading and Resource Allocation in Vehicle Edge Computing and Networks." IEEE Transactions on Vehicular Technology 68, no. 11 (2019): 11158-11168.

[6] Ning, Zhaolong, Peiran Dong, Xiaojie Wang, Joel JPC Rodrigues, and Feng Xia. "Deep Reinforcement Learning for Vehicular Edge Computing: An Intelligent Offloading System." ACM Transactions on Intelligent Systems and Technology (TIST) 10, no. 6 (2019): 1-24.

[7] He, Ying, Yuhang Wang, Chao Qiu, Qiuzhen Lin, Jianqiang Li, and Zhong Ming. "Blockchain-Based Edge Computing Resource Allocation in IoT: A Deep Reinforcement Learning Approach." IEEE Internet of Things Journal 8, no. 4 (2020): 2226-2237.

[8] Huang, Liang, Suzhi Bi, and Ying-Jun Angela Zhang. "Deep Reinforcement Learning for Online Computation Offloading in Wireless Powered Mobile-Edge Computing Networks." IEEE Transactions on Mobile Computing 19, no. 11 (2019): 2581-2593.

[9] Zheng, Tao, Jian Wan, Jilin Zhang, and Congfeng Jiang. "Deep Reinforcement Learning-Based Workload Scheduling for Edge Computing." Journal of Cloud Computing 11, no. 1 (2022): 3.

[10] Qiao, Guanhua, Supeng Leng, Sabita Maharjan, Yan Zhang, and Nirwan Ansari. "Deep Reinforcement Learning for Cooperative Content Caching in Vehicular Edge Computing and Networks." IEEE Internet of Things Journal 7, no. 1 (2019): 247-257.

[11] Li, Yunzhao, Feng Qi, Zhili Wang, Xiuming Yu, and Sujie Shao. "Distributed Edge Computing Offloading Algorithm Based on Deep Reinforcement Learning." IEEE Access 8 (2020): 85204-85215.

[12] Liu, Qian, Long Shi, Linlin Sun, Jun Li, Ming Ding, and Feng Shu. "Path Planning For UAV-Mounted Mobile Edge Computing with Deep Reinforcement Learning." IEEE Transactions on Vehicular Technology 69, no. 5 (2020): 5723-5728.

[13] Zhan, Wenhan, Chunbo Luo, Jin Wang, Chao Wang, Geyong Min, Hancong Duan, and Qingxin Zhu. "Deep-Reinforcement-Learning-Based Offloading Scheduling for Vehicular Edge Computing." IEEE Internet of Things Journal 7, no. 6 (2020): 5449-5465.

[14] Alfakih, Taha, Mohammad Mehedi Hassan, Abdu Gumaei, Claudio Savaglio, and Giancarlo Fortino. "Task Offloading and Resource Allocation for Mobile Edge Computing by Deep Reinforcement Learning Based on SARSA." Ieee Access 8 (2020): 54074-54084.

[15] Sheng, Shuran, Peng Chen, Zhimin Chen, Lenan Wu, and Yuxuan Yao. "Deep Reinforcement Learning-Based Task Scheduling in IoT Edge Computing." Sensors 21, no. 5 (2021): 1666.

[16] Wang, Liang, Kezhi Wang, Cunhua Pan, Wei Xu, Nauman Aslam, and Arumugam Nallanathan. "Deep Reinforcement Learning Based Dynamic Trajectory Control for UAV-Assisted Mobile Edge Computing." IEEE Transactions on Mobile Computing 21, no. 10 (2021): 3536-3550.

[17] Yamansavascilar, Baris, Ahmet Cihat Baktir, Cagatay Sonmez, Atay Ozgovde, and Cem Ersoy. "Deepedge: A Deep Reinforcement Learning Based Task Orchestrator for Edge Computing." IEEE Transactions on Network Science and Engineering 10, no. 1 (2022): 538-552.

[18] Luo, Quyuan, Changle Li, Tom H. Luan, and Weisong Shi. "Collaborative Data Scheduling for Vehicular Edge Computing Via Deep Reinforcement Learning." IEEE Internet of Things Journal 7, no. 10 (2020): 9637-9650.

[19] Peng, Haixia, and Xuemin Shen. "Deep Reinforcement Learning Based Resource Management for Multi-Access Edge Computing in Vehicular Networks." IEEE Transactions on Network Science and Engineering 7, no. 4 (2020): 2416-2428.

[20] Goudarzi, Mohammad, Marimuthu Palaniswami, and Rajkumar Buyya. "A Distributed Deep Reinforcement Learning Technique for Application Placement in Edge and Fog Computing Environments." IEEE Transactions on Mobile Computing 22, no. 5 (2021): 2491-2505.

[21] Zhang, Liang, Bijan Jabbari, and Nirwan Ansari. "Deep Reinforcement Learning Driven UAV-Assisted Edge Computing." IEEE Internet of Things Journal 9, no. 24 (2022): 25449-25459.

[22] Chen, Zhao, and Xiaodong Wang. "Decentralized Computation Offloading for Multi-User Mobile Edge Computing: A Deep Reinforcement Learning Approach." EURASIP Journal on Wireless Communications and Networking 2020, no. 1 (2020): 188.

[23] Lei, Lei, Huijuan Xu, Xiong Xiong, Kan Zheng, Wei Xiang, and Xianbin Wang. "Multiuser Resource Control with Deep Reinforcement Learning in IoT Edge Computing." IEEE Internet of Things Journal 6, no. 6 (2019): 10119-10133.

[24] Geng, Liwei, Hongbo Zhao, Jiayue Wang, Aryan Kaushik, Shuai Yuan, and Wenquan Feng. "Deep-Reinforcement-Learning-Based Distributed Computation Offloading in Vehicular Edge Computing Networks." IEEE Internet of Things Journal 10, no. 14 (2023): 12416-12433.

[25] Yu, Shuai, Xu Chen, Zhi Zhou, Xiaowen Gong, and Di Wu. "When Deep Reinforcement Learning Meets Federated Learning: Intelligent Multitimescale Resource Management for Multiaccess Edge Computing In 5G Ultradense Network." IEEE Internet of Things Journal 8, no. 4 (2020): 2238-2251.