ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 13, Issue 1, pp: (18-24), Month: April 2026 – September 2026, Available at: www.paperpublications.org

Rule-Based QoS Governors versus Built-in Constrained Reinforcement Learning for Wireless Resource Management: A Review and Design Perspective

Aouari Abdelhamid[1], Yuting Li[2]

[1]School of Computer Science and Artificial Intelligence, Hubei University of Technology, Wuhan, China

[2]School of Computer Science and Artificial Intelligence, Hubei University of Technology, Wuhan, China

DOI: https://doi.org/10.5281/zenodo.20024605

Published Date: 04-May-2026

Abstract: **Deep reinforcement learning (DRL) has become a promising approach for wireless resource management because of its ability to adapt scheduling, power control, and allocation decisions under dynamic network conditions. However, many DRL-based wireless solutions optimize performance through reward shaping alone, which can make Quality-of-Service (QoS) constraints difficult to interpret, tune, or guarantee in practical deployments. This paper presents a review and design perspective on constraint-handling mechanisms for learning-based wireless resource management, with a particular focus on the comparison between built-in constrained reinforcement learning methods and external rule-based QoS governors. We discuss four major approaches: reward-penalty design, constrained Markov decision process formulations, safety-layer or shielding methods, and runtime QoS governor mechanisms. The analysis highlights that while constrained RL methods provide a principled mathematical framework, they may introduce additional training complexity and sensitivity to hyperparameter selection. In contrast, rule-based QoS governors offer a practical and interpretable way to monitor service degradation, enforce conservative recovery actions, and support deployment around legacy schedulers such as proportional fair scheduling. Based on this comparison, the paper argues that external QoS governors can serve as a useful middle layer between fully heuristic scheduling and fully autonomous learning-based control. Finally, we identify open challenges related to multi-cell coordination, real-time inference, traffic generalization, O-RAN integration, and standardized benchmarking for safe DRL-based wireless systems.**

Keywords: **Deep reinforcement learning, wireless resource management, QoS constraints, safe reinforcement learning, proportional fair scheduling, runtime governor, O-RAN.**

1. INTRODUCTION

Wireless networks are becoming more complex because users require higher data rates, lower latency, stable service, and better energy efficiency. In LTE, 5G, and future wireless systems, radio resources must be allocated under changing channel conditions, user mobility, traffic demand, interference, and power limitations. Traditional scheduling methods, such as proportional fair scheduling, are still widely used because they are simple, reliable, and easy to implement [1]. However, these methods usually depend on fixed rules and manually selected parameters, which can limit their ability to adapt when network conditions change quickly.

Deep reinforcement learning has recently become a promising approach for wireless resource management because it can learn adaptive control policies through interaction with the environment [2], [3]. It has been studied for scheduling, power control, user association, and resource allocation [4], [5]. However, applying DRL to wireless networks also creates

Quality-of-Service challenges. In many studies, QoS requirements such as minimum data rate, delay, fairness, and reliability are handled only by adding penalty terms to the reward function. This is simple, but it does not always provide clear control over QoS violations. If the reward is not balanced properly, the agent may improve one objective, such as energy saving, while reducing user satisfaction or fairness.

This paper presents a review and design perspective on constraint handling in DRL-based wireless resource management. It compares reward-penalty methods, constrained reinforcement learning, safety-layer approaches, and rule-based QoS governors [6], [7]. The aim is not to propose a new algorithm, but to discuss practical ways to protect QoS when DRL is used in wireless systems. This paper focuses on representative constraint-handling approaches commonly discussed in DRL-based wireless resource management, including reward-penalty design, constrained reinforcement learning, safety-layer mechanisms, and external QoS governors, rather than providing an exhaustive survey of all DRL algorithms. In particular, the paper argues that rule-based QoS governors can provide a simple and interpretable middle-layer solution between traditional scheduling and fully autonomous DRL control. The rest of this paper is organized as follows: Section 2 introduces the background, Section 3 reviews constraint-handling methods, Section 4 discusses QoS governor design, Section 5 compares the approaches, Section 6 presents open challenges, and Section 7 concludes the paper.

2. BACKGROUND

2.1 Wireless Resource Management and Deep Reinforcement Learning

Wireless resource management is one of the main functions of modern mobile networks. It decides how limited resources, such as bandwidth, time slots, resource blocks, and transmit power, are shared among users. In LTE, 5G, and future wireless systems, this process must consider channel quality, user demand, mobility, interference, and fairness. Traditional scheduling methods, such as proportional fair scheduling, are widely used because they are simple, stable, and easy to implement. Proportional fair scheduling is especially important because it tries to balance system throughput and user fairness [1]. However, traditional methods usually rely on fixed rules and manually selected parameters, which may not be flexible enough when network conditions change quickly. Deep reinforcement learning has been introduced as a more adaptive approach for wireless resource management. In a DRL-based system, an agent observes the current network state, takes an action, and receives a reward based on the effect of that action [2]. The state may include information such as channel quality, traffic demand, queue length, previous throughput, or power usage. The action may represent scheduling decisions, power allocation, user association, or parameter adjustment [4], [5]. Recent work has also explored DRL-assisted proportional fair scheduling, where the learning controller adjusts selected scheduling-related parameters instead of replacing the whole scheduler [8]. Over time, the agent learns a policy that aims to improve long-term network performance. This makes DRL useful for dynamic wireless environments, where fixed scheduling rules may not always provide the best performance.

2.2 QoS Constraints and the Need for Safe Control

Although DRL is promising, its use in wireless networks is still challenging because network control decisions must satisfy Quality-of-Service requirements. QoS may include minimum data rate, low delay, low packet loss, stable connection, and fairness among users. These requirements are important because different network objectives often conflict with each other. For example, reducing transmit power can improve energy efficiency, but it may also reduce user data rates. Maximizing total throughput may improve system capacity, but it may reduce fairness if the scheduler mainly serves users with strong channel conditions. Therefore, a DRL agent should not only optimize performance, but also avoid actions that cause unacceptable service degradation. In many DRL-based wireless studies, QoS is handled by adding penalty terms to the reward function. This approach is simple and easy to implement, but it does not always provide clear control over constraint violations. If the reward terms are not balanced carefully, the agent may learn a policy that improves the numerical reward while still harming service quality. To reduce this risk, different constraint-handling methods have been studied, including constrained reinforcement learning, safety layers, shielding mechanisms, and external rule-based QoS governors [6], [7], [9]. Among these methods, rule-based governors are attractive for practical wireless systems because they are simple, interpretable, and can be added around existing schedulers without replacing the original scheduling algorithm.

3. CONSTRAINT-HANDLING METHODS IN DRL-BASED WIRELESS RESOURCE MANAGEMENT

3.1 Reward-Penalty and Constrained Reinforcement Learning Methods

One of the most common ways to handle QoS requirements in DRL-based wireless resource management is to include them directly in the reward function [4]-[6]. In this method, the reward is designed to encourage desirable behavior, such as high throughput, low delay, good fairness, and low power consumption. At the same time, the reward may include penalty terms when the agent violates QoS requirements, such as failing to meet a minimum data rate or using excessive transmit power. This approach is simple and widely used because it does not require major changes to the reinforcement learning algorithm. However, its main weakness is that the final behavior depends heavily on the design and scaling of the reward terms. If the penalty is too small, the agent may ignore QoS violations. If the penalty is too large, the agent may become too conservative and fail to optimize performance. Beyond reward penalties, a more formal approach is constrained reinforcement learning, where the wireless resource management problem is modeled with explicit constraints [7], [9]. In this case, the objective is not only to maximize the expected reward, but also to keep certain cost functions below acceptable limits. For example, the agent may try to maximize throughput while keeping delay, power consumption, or outage probability below predefined thresholds. This type of formulation is useful because it treats QoS as a constraint rather than only as a soft penalty. However, constrained reinforcement learning methods are often more difficult to train than standard DRL methods. They may require additional parameters, constraint estimators, or Lagrangian multipliers, which can make the learning process sensitive to tuning. In practical wireless networks, where channel conditions and traffic demand are highly dynamic, this complexity can make deployment more difficult.

3.2 Safety Layers, Shielding, and Rule-Based Governors

Another group of methods focuses on preventing unsafe or harmful actions before they affect the network [6], [9]. Safety layers and shielding mechanisms are examples of this approach. A safety layer checks the action selected by the DRL agent and modifies or blocks it if the action is expected to violate a constraint. For example, if an agent chooses a power allocation that exceeds a limit or causes severe QoS degradation, the safety layer can replace it with a safer action. This method can improve reliability because it adds protection outside the learning policy itself. However, it often requires a model or rule that can identify unsafe actions accurately. In complex wireless environments, designing such a safety mechanism can be difficult, especially when the effect of an action is uncertain or delayed.

Rule-based QoS governors follow a similar idea but are usually simpler and more interpretable. Instead of trying to predict every unsafe action directly, a governor monitors important network indicators during operation. These indicators may include average throughput, user satisfaction, delay, fairness, or power consumption. If the monitored performance drops below an acceptable level, the governor applies corrective rules to guide the system back to a safer state. For example, it may limit aggressive power reduction, restore conservative scheduling parameters, or gradually increase the resource budget until QoS improves. This makes the governor useful as an external protection layer around a DRL controller. Compared with fully built-in constrained RL methods, rule-based governors are less mathematically elegant but often easier to understand and deploy [6], [7], [9]. They are especially suitable for wireless systems where existing schedulers should not be replaced completely. In such systems, the DRL agent can be used to tune selected parameters or improve efficiency, while the governor ensures that the system does not move too far away from acceptable QoS behavior. This creates a practical balance between adaptability and safety. For this reason, rule-based QoS governors can be considered a useful middle-ground solution for applying DRL in realistic wireless resource management scenarios.

4. RULE-BASED QOS GOVERNORS FOR WIRELESS SCHEDULING

4.1 General Design Idea

A rule-based QoS governor is an external control mechanism that monitors the behavior of a wireless resource management system and intervenes when the service quality becomes worse than an acceptable level [6], [8], [9]. Unlike reward-penalty methods, the governor does not depend only on the internal learning objective of the DRL agent. Instead, it observes practical network indicators such as user satisfaction, average throughput, delay, fairness, and power usage. When these indicators remain within a safe range, the DRL controller is allowed to continue optimizing the system. However, when the indicators show a clear QoS degradation, the governor applies simple correction rules to push the system back toward safer operating conditions.

ISSN 2350-1022

International Journal of Recent Research in Mathematics Computer Science and Information Technology
Vol. 13, Issue 1, pp: (18-24), Month: April 2026 – September 2026, Available at: www.paperpublications.org

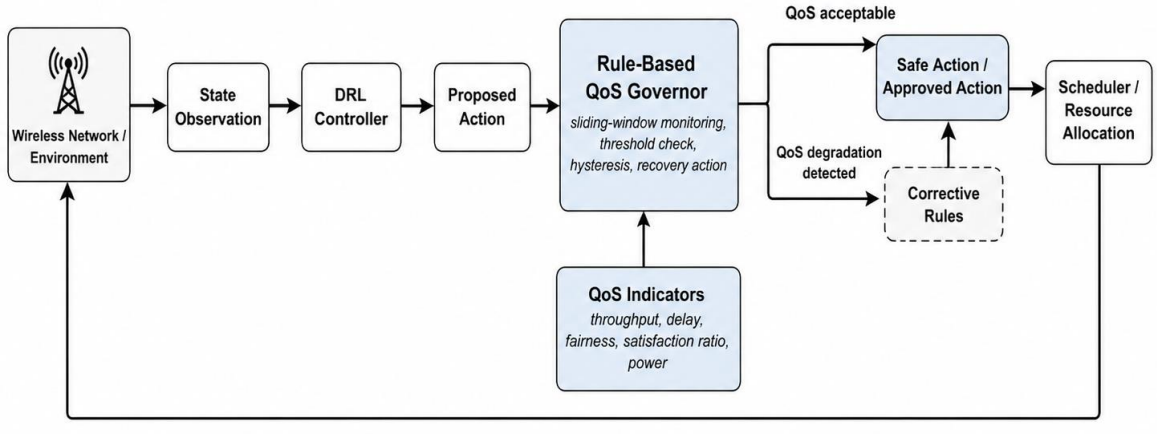

Figure 1: General structure of a DRL controller with a rule-based QoS governor

As shown in Fig. 1, this design is useful because it separates optimization from protection. The DRL agent can focus on improving long-term performance, such as reducing power consumption or adapting scheduling parameters, while the governor focuses on preventing unacceptable service degradation. In wireless scheduling, this is important because an agent may sometimes choose aggressive actions that look beneficial in terms of reward but reduce the quality experienced by some users. A governor provides an additional supervision layer that can limit these risky actions without completelydisabling the learning-based controller.

4.2 Monitoring and Intervention Mechanism

A practical QoS governor can monitor performance using a sliding time window instead of reacting to a single bad time slot [8]. This is important because wireless networks naturally experience short-term fluctuations due to fading, interference, and traffic variation. If the governor reacts too quickly, it may become unstable and interfere with normal scheduling behavior. By using a window-based measurement, the governor can detect persistent QoS degradation rather than temporary changes. For example, it can compare the average user satisfaction or average throughput over the recent window with a predefined reference value. A typical rule-based QoS governor can operate in four main steps. First, it collects recent QoS indicators such as satisfied-user ratio, average throughput, delay, fairness, or power usage. Second, it computes a sliding-window average to avoid reacting to short-term channel fluctuations. Third, it compares the measured QoS value with a predefined reference or threshold. Finally, if persistent degradation is detected, it applies a conservative correction, such as increasing the power budget, narrowing the action range, or returning the scheduler to safer parameters.

Procedure 1. Rule-Based QoS Governor

1. Observe recent network indicators such as throughput, delay, satisfaction ratio, fairness, and power usage.
2. Compute the sliding-window QoS value over the latest monitoring interval.
3. Compare the measured QoS value with the target threshold.
4. If QoS is acceptable, allow the DRL controller to continue normal optimization.
5. If QoS degradation is detected, restrict aggressive actions and move the system toward safer settings.
6. Return to normal operation only after QoS remains stable for a sufficient period.

Procedure 1 is intentionally simple so that the correction logic can be interpreted and adjusted by network operators.

When the monitored QoS falls below the acceptable threshold, the governor can apply conservative recovery actions. In a power-control or scheduling-parameter tuning system, this may include increasing the allowed transmit power, returning to safer scheduling parameters, or limiting how much the DRL agent can change the system at each step. Hysteresis can also be used to avoid frequent switching between normal and recovery modes [6], [8]. This means the governor does not immediately return to normal operation as soon as the QoS slightly improves. Instead, it waits until the performance becomes clearly stable again.

The main advantage of this approach is interpretability. The rules used by the governor are easy to understand and can be checked by network operators. For example, a rule such as "increase the power scale when the satisfied-user ratio remains below the target for several time steps" is easier to explain than a hidden neural network decision. This makes rule-based governors suitable for practical wireless systems, especially when the learning-based controller is added around an existing scheduler rather than replacing the scheduler itself. A rule-based governor also supports gradual deployment. Instead of giving the DRL agent full control over the scheduler, the system can first allow the agent to tune only a small number of parameters under strict governor protection. If the controller performs well, the allowed control range can later be expanded. This makes the approach safer and more realistic for wireless networks, where reliability and service stability are more important than achieving the highest possible reward in simulation.

5. COMPARISON OF CONSTRAINT-HANDLING APPROACHES

Different constraint-handling methods can be used when applying DRL to wireless resource management, but each method has different strengths and weaknesses. Reward-penalty methods are the simplest and most common approach [4]-[6]. They are easy to implement because QoS requirements are added directly into the reward function. However, they depend strongly on the choice of reward weights. If the weights are not selected carefully, the agent may focus too much on one objective and ignore another. For example, it may reduce power consumption while causing lower throughput or worse fairness. In comparison, constrained reinforcement learning methods provide a more formal way to handle QoS requirements [7], [9]. Instead of treating QoS only as part of the reward, these methods define constraints that the agent should satisfy during learning or operation. This can be useful for problems where delay, power, or outage probability must remain below a certain limit. However, these methods are usually more difficult to train and tune. They often require additional parameters, cost functions, or multiplier updates. In dynamic wireless environments, this can make the system more complex and less stable if the constraints are not designed properly.

Safety layers and shielding mechanisms offer another type of protection [6], [9]. They try to prevent unsafe actions before they are applied to the network. This can improve reliability because the DRL agent is not allowed to directly execute actions that violate important limits. However, these methods usually require a clear understanding of what makes an action unsafe. In wireless networks, this may not always be easy because the effect of an action can depend on channel variation, interference, user mobility, and traffic changes. As a result, safety layers can be effective, but they may also be difficult to design for complex real-world systems. Rule-based QoS governors follow the same general safety idea, but they provide a simpler and more interpretable alternative [6], [8], [9]. Instead of changing the reinforcement learning algorithm itself, the governor monitors the system during operation and applies correction rules when QoS becomes poor. This makes it easier to understand and easier to deploy around existing wireless schedulers. The main limitation is that rule-based governors may be conservative. They may reduce the freedom of the DRL agent and prevent it from exploring more aggressive but potentially useful actions. However, for practical wireless systems, this conservativeness can be acceptable because service stability is often more important than maximum theoretical performance.

Overall, no single method is best for all situations. Reward penalties are useful for simple simulations, constrained RL is useful when a formal mathematical constraint formulation is needed, safety layers are useful when unsafe actions can be clearly identified, and rule-based governors are useful when interpretability and deployment feasibility are important. For practical wireless scheduling, a rule-based QoS governor can be a strong option because it can protect service quality while still allowing a DRL controller to improve efficiency within safe limits.

Table I summarizes the main differences between the reviewed constraint-handling approaches.

Table I: Comparison of Constraint-Handling Methods

Method	Main Advantage	Main Limitation	Suitable Use
Reward-penalty design	Simple and easy to implement	Sensitive to reward weight selection	Early DRL simulations
Constrained reinforcement learning	More formal constraint modeling	More complex training and tuning	Research problems with clear constraints
Safety layer / shielding	Blocks unsafe actions before execution	Requires accurate safety rules or models	Systems where unsafe actions can be identified
Rule-based QoS governor	Interpretable and practical	Can be conservative	Deployment around existing schedulers

6. OPEN CHALLENGES AND FUTURE RESEARCH DIRECTIONS

Although DRL-based wireless resource management has shown strong potential, several challenges still need to be addressed before it can be widely used in practical networks. One important challenge is generalization [10]. A DRL model trained under one network condition may not perform well when the number of users, traffic demand, channel quality, or interference pattern changes. Wireless environments are highly dynamic, so a controller must be able to maintain stable performance under different scenarios rather than only working well in one simulation setting. Another challenge is real-time deployment [11], [12]. Wireless scheduling decisions often need to be made within a very short time, especially in LTE and 5G systems. If a DRL model is too complex, its inference time may be too high for practical use. This means that future research should not only focus on improving reward or energy efficiency, but also on reducing computational complexity. Lightweight models, slower-timescale control, and parameter-tuning approaches may be more practical than replacing the full scheduler with a neural network.

QoS protection also remains an open issue. Reward penalties, constrained reinforcement learning, safety layers, and rule-based governors each have advantages, but none of them fully solves the problem. Reward-based methods are simple but can be unstable if the reward is not carefully balanced. Constrained reinforcement learning is more formal but harder to train. Rule-based governors are interpretable but may be conservative. Future systems may combine these methods, for example by using constrained RL during training and a rule-based governor during deployment [7], [9]. Another important direction is integration with existing network architectures. In practical networks, operators may not want to replace trusted scheduling methods completely. Therefore, DRL may be more acceptable if it is used as an assistant controller that tunes selected parameters while the original scheduler remains active. This idea is also related to the development of intelligent and open radio access network architectures, where AI-based controllers can support optimization without directly taking full control of all network functions [13], [14].

Finally, fair evaluation and benchmarking are needed [4], [10], [15]. Many DRL-based wireless studies use different environments, reward functions, metrics, and traffic assumptions, which makes comparison difficult. Future work should use clearer benchmarks and report multiple performance indicators, including throughput, fairness, delay, power consumption, satisfaction ratio, and constraint violation rate. This would make it easier to understand whether a proposed method is truly practical or only performs well under limited simulation conditions.

7. CONCLUSION

Deep reinforcement learning is a promising tool for wireless resource management because it can adapt to changing network conditions and improve decisions related to scheduling, power control, and resource allocation. However, its practical use is still limited by the difficulty of protecting Quality-of-Service requirements. If QoS is handled only through reward penalties, the agent may learn behavior that improves the reward but still causes service degradation. This makes constraint handling an important part of any DRL-based wireless system.

This paper reviewed several approaches for handling constraints in DRL-based wireless resource management, including reward-penalty design, constrained reinforcement learning, safety layers, shielding mechanisms, and rule-based QoS governors. Each approach has its own advantages and limitations. Reward penalties are simple, but sensitive to weight selection. Constrained reinforcement learning is more formal, but harder to train and tune. Safety layers can block harmful actions, but may require accurate models of unsafe behavior. Rule-based QoS governors are simpler and more interpretable, making them suitable for practical systems where stability and reliability are important.

The main argument of this paper is that rule-based QoS governors can provide a useful middle-layer solution between traditional wireless scheduling and fully autonomous DRL control. Instead of replacing existing schedulers, a DRL controller can be used to tune selected parameters, while the QoS governor monitors performance and prevents unacceptable service degradation. This approach can make learning-based wireless optimization safer, easier to explain, and more compatible with real network deployment. Future research should focus on combining learning-based optimization with interpretable protection mechanisms, improving generalization across different network conditions, and developing fair benchmarks for safe DRL-based wireless resource management.

REFERENCES

[1] H. J. Kushner and P. A. Whiting, "Convergence of proportional-fair sharing algorithms under general conditions," IEEE Transactions on Wireless Communications, vol. 3, no. 4, pp. 1250–1259, 2004.

[2] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction, 2nd ed. Cambridge, MA, USA: MIT Press, 2018.

[3] T. Haarnoja, A. Zhou, K. Hartikainen, G. Tucker, S. Ha, J. Tan, V. Kumar, H. Zhu, A. Gupta, P. Abbeel, and S. Levine, "Soft Actor-Critic algorithms and applications," arXiv preprint arXiv:1812.05905, 2018.

[4] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues," IEEE Communications Surveys & Tutorials, vol. 21, no. 4, pp. 3072–3108, 2019.

[5] J. A. Hurtado Sánchez, K. Casilimas, and O. M. Caicedo Rendon, "Deep reinforcement learning for resource management on network slicing: A survey," Sensors, vol. 22, no. 8, Article 3031, 2022.

[6] J. García and F. Fernández, "A comprehensive survey on safe reinforcement learning," Journal of Machine Learning Research, vol. 16, pp. 1437–1480, 2015.

[7] J. Achiam, D. Held, A. Tamar, and P. Abbeel, "Constrained policy optimization," in Proceedings of the 34th International Conference on Machine Learning (ICML), 2017, pp. 22–31.

[8] A. Aouari, Y. Chen, and Y. Li, "Deep reinforcement learning-based proportional fair scheduling for power-efficient LTE/5G downlink," in Proceedings of the 2026 International Conference on Communication Networks and Machine Learning (CNML), 2026, doi: 10.1109/CNML68938.2026.11452406.

[9] S. Gu, L. Yang, Y. Du, G. Chen, F. Walter, J. Wang, and A. Knoll, "A review of safe reinforcement learning: Methods, theory and applications," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 46, no. 12, pp. 11216–11235, 2024.

[10] B. Demirel, Y. Wang, C. Tatino, and P. Soldati, "Generalization in reinforcement learning for radio access networks," arXiv preprint arXiv:2507.06602, 2025.

[11] A. Vetsos et al., "Comparative analysis of open-source 5G simulators with SDAP/QoS flow support," ACM SIGAPP Applied Computing Review, vol. 24, no. 2, pp. 44–54, 2024.

[12] O-RAN Alliance, "O-RAN Architecture Description," O-RAN technical specification, latest available release.

[13] M. Martínez-Morfa, C. Ruiz de Mendoza, C. Cervelló-Pastor, and S. Sallent, "DRL-based xApps for dynamic RAN and MEC resource allocation and slicing in O-RAN," in Proceedings of the 15th International Conference on Network of the Future (NoF), 2024, pp. 106–114.

[14] A. H. Mohammed, S. T. Shah, Y. A. Sambo, and M. Imran, "From concept to reality: QoS-based RAN slicing in next generation O-RAN networks," in Proceedings of IEEE CAMAD, 2024, pp. 1–7.

[15] M. Zangooei, N. Saha, M. Golkarifard, and R. Boutaba, "Reinforcement learning for radio resource management in RAN slicing: A survey," IEEE Communications Magazine, vol. 61, no. 2, pp. 118–124, 2023.