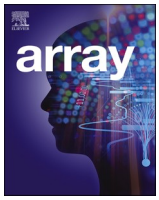
Array 14 (2022) 100142

Contents lists available at ScienceDirect

Array

journal homepage: www.sciencedirect.com/journal/array



A survey of deep reinforcement learning application in 5G and beyond network slicing and virtualization

Charles Ssengonzi [a], Okuthe P. Kogeda [a], Thomas O. Olwal [b,*]

[a] University of the Free State, Faculty of Natural and Agricultural Sciences, Department of Computer Science and Informatics, P. O. Box 339, Bloemfontein, 9300, South Africa

[b] Tshwane University of Technology, Faculty of Engineering and the Built Environment, Department of Electrical Engineering, Pretoria, South Africa

ARTICLE INFO

Keywords:
Machine learning
Reinforcement learning
Deep reinforcement learning
5G
Multi-domain network slicing
Orchestration
Admission control
Prediction

ABSTRACT

The 5th Generation (5G) and beyond networks are expected to offer huge throughputs, connect large number of devices, support low latency and large numbers of business services. To realize this vision, there is a need for a paradigm shift in the way cellular networks are designed, built, and maintained. Network slicing divides the physical network infrastructure into multiple virtual networks to support diverse business services, enterprise applications and use cases. Multiple services and use cases with varying architectures and quality of service requirements on such shared infrastructure complicates the network environment. Moreover, the dynamic and heterogeneous nature of 5G and beyond networks will exacerbate network management and operations complexity. Inspired by the successful application of machine learning tools in solving complex mobile network decision making problems, deep reinforcement learning (Deep RL) methods provide potential solutions to address slice lifecycle management and operation challenges in 5G and beyond networks. This paper aims to bridge the gap between Deep RL and the 5G network slicing research, by presenting a comprehensive survey of their existing research association. First, the basic concepts of Deep RL framework are presented. 5G network slicing and virtualization principles are then discussed. Thirdly, we review challenges in 5G network slicing and the current research efforts to incorporate Deep RL in addressing them. Lastly, we present open research problems and directions for future research.

1. Introduction

The vision of 5G in Ref. [1] is to support a variety of applications and uses cases classified as enhanced mobile broadband (eMBB), ultra-reliable low-latency communications (uRLLC) and massive machine-type communications (mMTC). Example of use cases include smart homing, autonomous driving, intelligent transportation, and many others [2] with key technology enablers being software defined networking (SDN) [3], network function virtualization (NFV) [4], mobile edge computing (MEC) [5], and cloud computing [6]. The use cases feature varying radio access network (RAN) architectures, features, and quality of service (QoS) requirements. The need to simplify network operations will therefore necessitate new approaches to network design, construction, and maintenance. Network slicing [9] has emerged as potential solution to address these challenges. It transforms the network build philosophy of the previous generations' monolithic architecture to the creation and operation of virtual/logical networks tailored to the specific needs of specific customers, and services, within the same physical network infrastructure. Flexible network operation and management will be realized through the creation, modification and deletion of slices whenever needed [9] The slices are isolated, and each can be assigned to a use case with specific QoS requirements to guarantee service level agreements (SLAs) [9]. The network resources can be allocated to each slice on demand and based on precise requirements to prevent over provisioning [9]. Several 5G network slicing pilots in Ref. [12] showed promising results. However, they face challenges in (a) slicing across multiple domains, (b) joint multi-dimensional network resource orchestration and (c) striking a balance between maintaining QoS/QoE and maximizing network utility (d) the need for a priori traffic profiles in a dynamic environment.

Deep RL's excellent performance in mobile networking [14,21,22] is the basis for its choice as an alternative to statistical model based solutions to address 5G slicing challenges. Deep RL is a combination of RL and Deep neural networks (DNN). RL interacts with the environment in

* Corresponding author.
E-mail addresses: charles.ssengonzi@ericsson.com (C. Ssengonzi), kogedapo@ufs.ac.za (O.P. Kogeda), olwalto@tut.ac.za (T.O. Olwal).

https://doi.org/10.1016/j.array.2022.100142
Received 3 January 2022; Accepted 29 March 2022
Available online 4 April 2022

a trial and error manner [35], learning from its actions to improve rewards. That way, it resolves the need for a priori traffic profiles and addresses InP utility maximization challenges. However, it often performs poorly in large state-action spaces, referred to as the curse of dimensionality. When combined with DNNs, this challenge can be addressed. That way it addresses feature extraction and optimal policy search challenges expected in multi-domain and joint multi-dimensional resource allocation scenarios. Deep RL's popularity continues to grow in the 5G networking domain. However, existing literature is scattered across different research areas and there is significant shortage of studies that comprehensively address its use in the 5G multi-domain network slicing scenarios. This paper bridges this gap and presents a detailed survey of the association between Deep RL, Network slicing and Virtualization in 5G and beyond networks. This paper's significant contributions include:

- An extensive survey of 5G network slicing and virtualization from an RL and Deep RL angle. To the best of our knowledge, this is the first paper in that regard.
- Deep RL and its motivation for use in 5G mobile communication scenarios.
- A review of current studies to automate 5G slice lifecycle management using Deep RL.
- Open research problems and directions for further investigation.

The remainder of this paper is organized as follows: Section 2 reviews existing surveys related to the topic. Section 3 introduces RL and Deep RL. Section 4 provides an over of 5G networks. The foundations and principles of 5G network slicing and virtualization are introduced in Section 5. Section 6 introduces Management and Orchestration. Section 7 reviews exiting studies towards the use of Deep RL in 5G network slicing, mainly focusing on (i) Network Topology, (ii) Admission Control, (iii) Resource allocation and Management and (iv) Traffic Forecast and Prediction areas critical to 5G Infrastructure providers (InPs) and slice tenants. Lastly, open challenges and directions for further investigation are presented in Section 8. A list of important abbreviations and definitions related to the topic are given in Table 1. A summary of each of the sections together with the lessons learnt will also be presented at the end of each review section. A summary of existing surveys, magazines, papers, and books related to RL/Deep RL framework, Network slicing and virtualization is presented in Table 2.

Table 1
A list of important abbreviations related to the study.

Abbreviation	Full description
5G	5th Generation network
AI	Artificial Intelligence
D2D	Device to Device
Deep RL	Deep Reinforcement Learning
DNN	Deep Neural Network
DQL	Deep-Q Learning
DQN	Deep Q-Network
IaaS	Infrastructure- as- a-service
MDP	Markov Decision Process
MEC	Mobile Edge Computing
ML	Machine Learning
NSSF	Network Slice Selection Function
NSSI	Network Slice Subnet Instance
PaaS	Platform- as- a-Service
POMDP	Partially Observable MDP
QoE	Quality of Experience
QoS	Quality of Services
RAN	Radio Access Network
RL	Reinforcement Learning
SaaS	Software-*as*-a-Service
SaaS	Slice as a service
SLA	Service Level Agreement
TCO	Total Cost of Ownership
URLLC	Ultra-Reliable Low Latency Communications
V2X	Vehicle to X
IaaS	Infrastructure as a service
PaaS	Platform as a service

2. A review of survey papers related to the topic

The need to address 5G network complexity, monetization challenges and the huge operation and management expenditure, has accelerated research in multi-domain network slicing and AI/ML tools such as Deep RL. However, several existing studies show that these two topics continue to be studied individually. Current efforts investigating these two topics can be classified into the following:

- Surveys on the RL framework and its applications.
- Surveys on the Deep RL framework and its applications.
- Surveys on the Deep RL in 5G mobile communication networks. Network slicing is often mentioned in a few paragraphs here.
- Surveys on 5G network slicing and virtualization.
- Study tutorials, books, papers, and magazines on RL, Deep RL, and network slicing.

In this section we review existing survey papers related to our topic. A diagrammatic view of the organization of this survey is presented in Fig. 1.

2.1. Surveys on RL and deep RL framework and its applications

Reinforcement learning (RL) is an ML approach for tackling sequential decision making problems based on Markov decision processes [28]. The research in RL has been dramatically accelerated by its strong foundations, breakthroughs, and excellent performance of Google DeepMind's AlphaGo and Alphazero in Go [17,19] games. For example, the authors in Ref. [29] provide a comprehensive overview of the historical foundations of the RL domain and the core issues. The study in Ref. [30] investigates the role of the Bayesian method in the RL paradigm, the detailed Bayesian RL algorithm, and their theoretical and empirical properties. The study in Ref. [31] provides a comprehensive overview of the RL and Deep RL methods and their application in economics. Inverse Reinforcement Learning (IRL) is considered in Ref. [32]. IRL uses data generated during the execution of a task to build an autonomous agent that can model other agents without impacting the performance of the task. Preference-based reinforcement learning (PbRL), which learns from non-numerical rewards, is reviewed in Ref. [33].

Applications of RL in 5G wireless communications is surveyed in Ref. [34]. GAN-powered deep distributional RL [39], safe RL [38], transfer learning [40], multiagent RL [41], Optimal autonomous control using RL [42], model based RL [43] and RL in dynamically varying environments [44] are other existing surveys related to this paper. Complimenting RL with deep neural networks (DNN) pioneered the Deep RL field and its crucial breakthroughs in various disciplines and domains. The authors in Refs. [45,46] reviewed the application of Deep RL in automation and robotics, preordained to revolutionize the manufacturing industry and reduce the total cost of ownership (TCO). Deep RL application in mobile communication was surveyed in Ref. [47]. The authors in Ref. [48] provide a comprehensive review and critique of the state-of-art in multi agent Deep RL. The authors in Ref. [49] reviewed Deep RL efforts devoted to cyber-physical systems, autonomous intrusion detection techniques and defence strategies against cyber-attacks, creating the potential to revolutionize cyber security mechanism. The authors in Ref. [50] surveyed DRL efforts in online advertising and recommender systems, which is intended to improve mobile advertising. Recent surveys [51–53] focus on Deep RL approaches in real world autonomous driving and video gaming.

Table 2
Summary of existing surveys, magazines, papers, and books related to RL/Deep RL framework, Network slicing and virtualization. ■ implies that the publication is in scope of a domain and directly linked to the topic. × implies that whereas important insights can be obtained from that publication, it does not directly cover that area.

Overview		Enablers			ML Tools				Networks		
Author	Description	MEC	NFV	SDN	ML	DL	RL	DRL	NS	5G	6G
Ordonez et al. [4]	Network slicing	■	■	■					■	■	
Sutton et.al [35]	RL Textbook				×	×	■	■			
Kaelbling et al. [29]	RL intellectual Foundations				×	×	■	■			
Mosavi et al. [31]	RL, Deep RL in economics				×	×	■	■			
Wirth et al. [33]	Preference-based RL				×	×	■				
Ghavamzadeh et al. [30]	Bayesian methods in RL				×	×	■				
Arora et al. [32]	Inverse RL				×	×	■				
Qian et al. [34]	RL and Deep RL in MEC	■			×	×	■	■		×	
Hua et al. [39]	GAN-powered distributional RL				×	■	■	■			
Garcia et al. [38]	Safe RL				×	×	■				
Moerland et al. [43]	Model-based RL				×	×	■				
Padakandla et al. [44]	RL algorithms for dynamic environments.				×	×	■				
Taylor et al. [40]	Transfer learning				×	×	■				
Busoniu et al. [41]	Multiagent RL				×	×	■				
Kiumarsi et al. [42]	autonomous control				×	×	■				
Gguyen et al. [45]	Deep RL methods in automation and robotics				×	×	■	■			
Li et al. [47]	Deep RL Tutorial				×	×	■	■			
Xiong et al. [48]	Multi Agent RL						■				

Continuation from the previous page above											
Zhao et al. [50]	Deep RL methods in automation and robotics				×	×				×	×
Yu et al. [51]	Deep RL for Smart Building Energy Management				×	×				×	
Kiran et al. [52]	Deep RL for autonomous systems				×	×				×	
Liang et al. [62]	5G NS architectural design										
Foukas et al. [59]	5G network slicing architecture										
Barakabitze et al. [63]	Network slicing review										
Khan et al. [64]	Network slicing review										
Feriani et al. [70]	Deep RL tutorial				×	×					
Jovovic et al. [60]	Service Function chaining										
Xiong et al. [55]	AI and ML in B5G networks					×					
Luong et al. [56]	Deep RL in networking				×	×					
Qian et al. [57]	Deep RL in MEC, SDN, NFV				×	×					
Our survey	Deep RL in 5G and Beyond network slicing and virtualization										

2.2. Surveys on deep RL in 5G mobile communication networks

As Deep RL continues to gain interest in the research community, there is an ever-growing number of publications on its applications in wireless networks. Recent research on cognitive, autonomous, intelligent 5G networks and beyond is summarized in Ref. [54]. The authors in Refs. [45,55] review AI and ML applications in the design and operation of 5G and beyond networks. The authors in Ref. [56] outline research into the application of single-agent Deep RL algorithms to solve mobile network problems such as network access control, data rate control, wireless caching, data offloading, and network security. The authors in Ref. [57] review the latest studies on Deep RL driven MEC, SDN, and NFV networks, the key technology enablers of 5G network slicing and virtualization. In Ref. [58], Deep RL's specific applications in the areas of the Internet of Things (IoT), resource management, and mobile edge caching domains are surveyed.

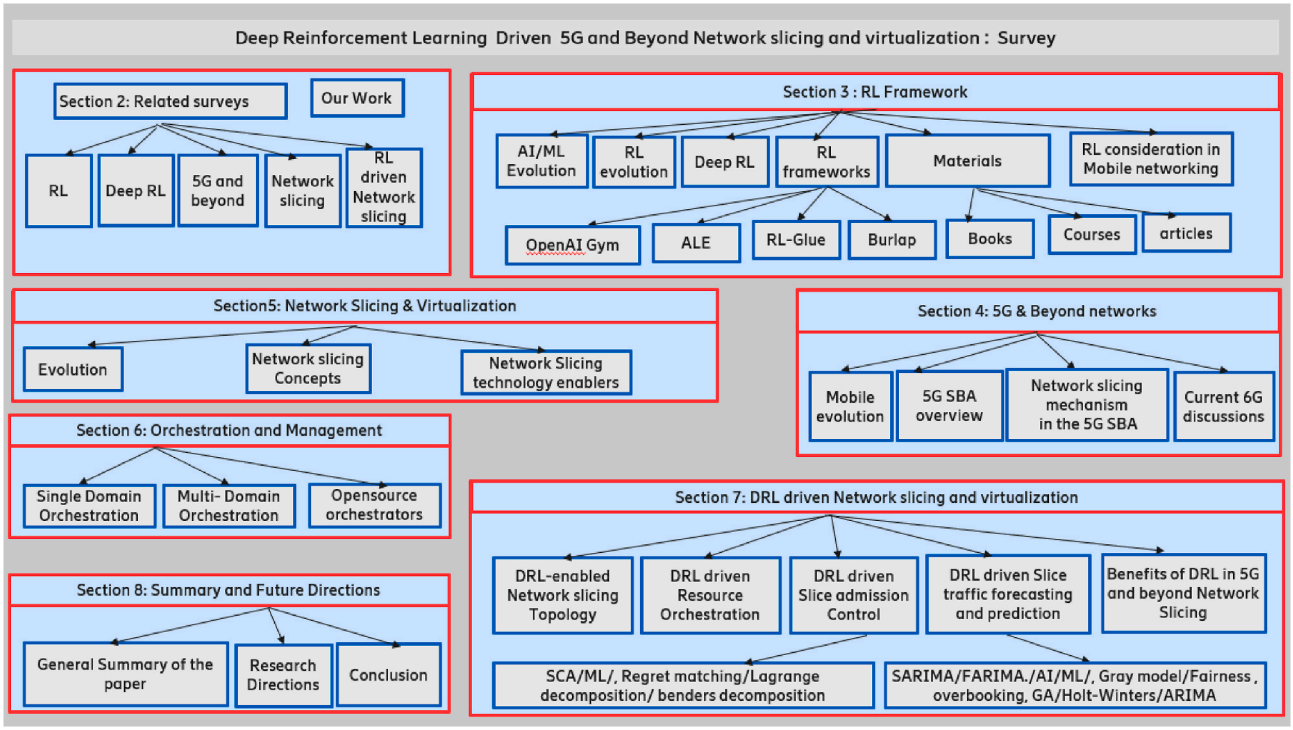


Fig. 1. A diagrammatic representation of the survey organization.

2.3. Surveys on 5G network slicing and virtualization

Several attempts have been made to review progress on 5G network slicing and provide insights into the standardization progress. The authors of [59] review early research on the general reference architecture of 5G network slicing, which consists of the infrastructure layer, network function layer, service layer, management and orchestration layer. The study in Ref. [60] focuses on network slicing principles, including dynamic service chaining, orchestration, and management. The authors of [62] review research in the architectural and topology aspects of 5G network slicing and virtualization. The authors of [64] outline the latest breakthroughs, and challenges in network slicing and virtualization. An overview of the management and orchestration (MANO) architecture across multiple domains is found in Ref. [65]. An overview of 5G network slice security is given in Ref. [67]. A comprehensive study of E2E multi-domain network slicing and its key enablers and use cases is presented in Ref. [63].

2.4. Fascinating RL, deep RL research materials

A comprehensive bibliometric assessment of worldwide RL publications from 2009 to 2018 can be found in Ref. [36]. The authors of [70,71], and [72] provide comprehensive tutorials on RL and Deep RL. The recently published tutorial in Ref. [63] provides a comprehensive overview of 5G network slicing and virtualization, its technical requirements, and information on 3GPP standardization efforts. A comprehensive summary of useful resources and resources specific to the RL and Deep RL frameworks is summarized in Section 3.6.

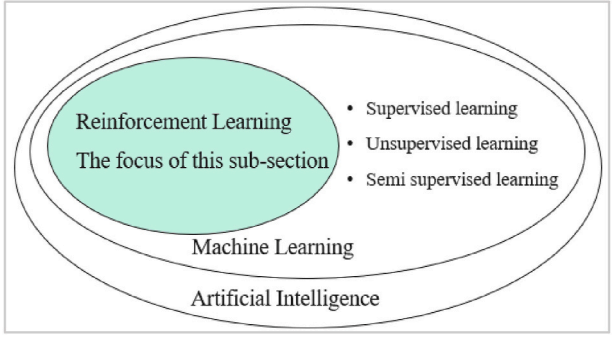


Fig. 2. Relationship between AI, ML, RL, SL, SSL, and USL.

2.5. Scope and objectives of this study

It's evident from the studies above that there is a paucity of literature tackling the association between the Deep RL and 5G network slicing research. Our goal is to bridge this gap. As a result, the following research questions are addressed in this paper:

a) What are the technology enablers and challenges in 5G and beyond network slicing?

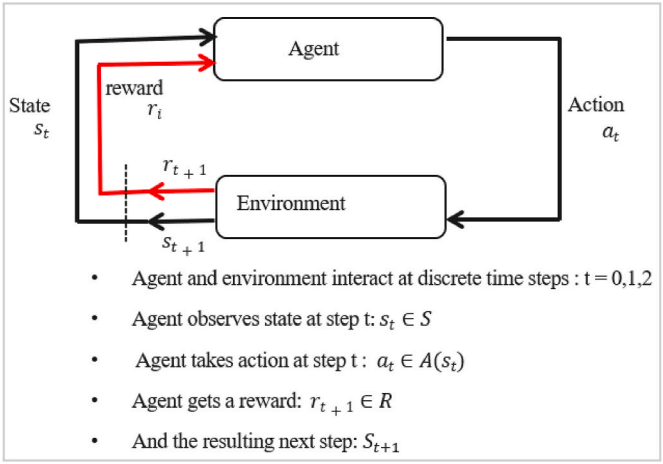


Fig. 3. Reinforcement learning process [35].

b) Why is Deep RL so promising for solving 5G and beyond network slicing challenges?
c) What are the successful Deep RL applications in 5G and beyond network slicing?
d) What are the promising directions in Deep RL and 5G network slicing for further study?

The existing review papers outlined above already addressed some of our research questions. However, our paper goes beyond these previous studies, with a particular focus on research associations between Deep RL and 5G network slicing. Our paper differs from existing studies in the following ways: (a) rather than focusing on Deep RL uses in the high-level 5G network slicing context as presented in Refs. [48,56], we focus on details surrounding its application in key network slicing functionalities, critical to InPs and slice tenants such as slice topology, resource allocation, admission control, forecasting and prediction.

(b) We review major RL and Deep RL platforms that can be used to develop and train RL and Deep RL agents, including environments that support testing agents that solve mobile network communication problems. (c) We review recent advances in Deep RL research and introduce readers to recent cutting-edge technologies that help researchers improve their RL/Deep RL agent training. (d) This paper focuses on Deep RL in network slicing for 5G networks and beyond. However, relevant to this paper, we also discuss the potential use of Deep RL in wireless communication domains in general. (e) We outline useful tutorials, books, papers, and other research materials, as well opensource management and orchestration platforms that will enable readers gain meaningful insights in the intellectual foundations of RL/Deep RL and at the same time quickly learn how to develop AI agents that operate freely in a 5G network slicing environment. The discussions will be summarized in tabular format. Lessons learned in each section will also be provided.

3. Overview of RL and deep RL

As a precursor to our study, we briefly introduce fundamental concepts of AI, RL, Deep RL, and some useful resources beneficial to researchers and academicians interested in this field of learning.

3.1. The agent and its environment

In their famous book, Russel and Norvig [73] define AI as "the art of creating machines that perform functions that require intelligence when performed by humans. The field of AI aims to understand and build useful and smart agents. An agent is defined as anything that recognizes its environment through sensors and acts on it through effectors [73]. The agents are autonomous because their own experience dictates their behaviour [73]. Consequently, their ability to operate successfully in a wide variety of environments should enable their integration in heterogeneous and dynamic 5G network environments plausible and interesting! No wonder, given the complexities and problems in end to end (E2E) 5G multi-domain network slicing, AI tools and methods provide alternative operation and maintenance solutions. AI has been widely used in many fields and domains to address massive and sophisticated cognitive problems that exceed the capabilities of humans.

$$V^{\pi}(\hat{s}) = E_{\pi}\left[\sum_{k=0}^{\infty}\gamma^{k} R\left(s^{(k)}, \pi\left(s^{(k)}\right)\right) \middle| s^{(0)} = \hat{s}\right]$$

$$= E_{\pi}\left[R(\hat{s}, \pi(\hat{s})) + \gamma \sum_{s' \in S} P\left(s' \middle| \hat{s}, \pi(\hat{s})\right) V^{\pi}(s')\right]$$

Fig. 4. Bellman's equation [89].

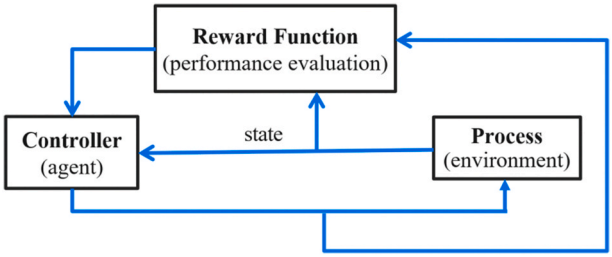


Fig. 5. The basic elements of DP and RL, and their flow of interaction in [89].

Machine learning (ML), a subfield of AI, has continued to break into new frontiers , exceeding human expectations.

3.2. Historical evolution

Machine learning (ML) tools and methods are often classified into (i) supervised learning, (ii) semi supervised learning, (iii) unsupervised learning, and (iv) RL [74]. Supervised learning (SL) necessitates the training of models based on labeled data. Unsupervised learning (USL) necessitates model training using unlabeled data. Semi supervised learning (SSL) requires a mixture of both labeled and unlabeled data. In RL, the learning takes place through continuous interaction with the environment in a trial and error manner [35]. Fig. 2 provides a summary of the relationship between SL, USL, SSL, RL, ML and AI.

- 1959: Arthur Samuel's work [75] was one the first successful investigation of ML. His work contained most of the modern ideas in RL, including temporal differentiation and input generalization.
- 1949 to the 1960s: Adaptive control theory researchers (Widrow and Hoff [76]), extended the work of Hebb in Refs. [77,78] and saw the application of least mean square (LSM) algorithms in RL agent training. The famous Bellman's equation and the 'curse of dimensionality"; concept was introduced in Bellman's work in Ref. [89].
- 1968: Michie and Chambers' work on adaptative control in Ref. [79] experimented on balancing Cartpole, exemplifying basic RL methods.
- 1980s: The work of Barto and Sutton in Refs. [80,81] is credited with revitalizing modern research in RL and AI/ML in general.
- Late 1980s and early 1990s: Fundamental reviews and descriptions of RL are presented in both Kaelbling's work in Ref. [83] and Sutton's work in Ref. [84]. Watkins doctoral thesis in Ref. [82] pioneered the Q-learning algorithm. Sutton's book [35] describes some critical foundations in RL. Koza's work in Ref. [88] pioneered genetic programming in the design of complex agents with mutation techniques that can be used to solve problems in several fields and domains including applied mathematics, control engineering, and AI.
- 2000s: Research on the RL gradually advanced but the application of RL was limited to domains where useful features could be extracted manually, or domains with a fully observed low-dimensional state space. With time, the introduction of neural networks for functional approximation addressed RL's limitations in higher order spaces. DeepMind's breakthroughs in the GO games, and the recent AlphaFold, RGB-Stacking publications have exponentially accelerated RL's research into other domains and frontiers. Other players including Microsoft, Google, Facebook, and Amazon continue to invest heavily in RL research and development.

3.3. RL fundamental principles

While SL and USL generate predictions and classifications based on labeled or unlabeled data, the RL agent interacts with its given environment, iteratively collecting data (experiences), and receiving a reward based on the action taken. A summary of the RL learning process

is shown in Fig. 3.

In essence, the RL agent monitors the state (s_t) of the environment in discrete time steps t and then takes an action (a_t) based on the policy $\pi(a/s)$. The environment reacts and the agent receives a reward (r_{t+1}) and the next state (s_{t+1}). The experiences associated with (s, a, r, s_{t+1}) represent the data that the agent uses to train the policy. The agent uses the updated state and reward to select the next action. This loop is used in episodes to learn how to get the most out of each one and repeated until the environment is terminated. The feedback can come directly from the environment, for example, a numerical counter in a visual environment, or as the result of a calculation or function. The goal of the RL agent is to learn the optimal behaviour or action policy at each state transition to maximize the reward. More detailed explanations on the foundations, principles, and concepts of RL can be found in Ref. [35].

3.3.1. RL problem formulation

To maximize the cumulative or average reward, the RL agent learns how to interact with the environment through a trial and error manner [35]. Markov Decision Process (MDP) represents this sequential decision making interaction using 5-tuples as $M = (S, A, P(s'/s, a), R, \gamma)$, where S and A denote a finite state and action set respectively, and $P(s'/s, a)$ indicates the probability that the action $a \in A$ under state $s \in S$ at time slot t leads to state $s' \in S$ at time slot $t+1$. The R often represents a reward and, in this case, $R(s, a)$ is an immediate reward after performing action a in state s. On the other hand, $\gamma \in [0, 1]$ is a discount factor to mirror the diminishing significance of current rewards on the future rewards. The MDP aims to find a policy $\pi(a/s)$ that determines the selected action a under state s, to maximize the value function as expressed in the Bellman's equation in Fig. 4, typically the expected discounted cumulative reward.

The state transitions are generally non-linear and sometimes probabilistic. This interaction pattern is shown in Fig. 5. If the state transition probability $P(s'/s, a)$ is known in advance without random variables, dynamic programming [89] methods are used to solve the Bellman problem. Since RL aims to obtain the optimal policy π* under circumstances with unknown and partially random dynamics, it's a candidate for use in heterogeneous and dynamic network environments such as 5G and beyond networks.

Classic RL algorithms include Q-learning [86] and its variants, actor-critic methods [91] and its variants, SARSA [92], TD (λ) [93] and many more. RL methods work in scenarios where the system model may or not exist. If the system model is available, dynamic programming methods such as policy evaluation can be used to calculate the value function for a policy and use value iteration and policy iteration to find the optimal policy [89]. An RL environment can be multi-armed bandit, an MDP, a partially observable MDP (POMDP), or a game.

3.3.2. Value function

Value functions indicate how good an agent is in a particular state, or how good a state action pair is. The value function approach is based on assessing the expected return and then trying to find the best strategy or action that maximizes the expected value in all the possible state-actions. The policy may be improved by iteratively assessing and updating the estimate of the value function. For a particular policy π, the state-value function $v_\pi(s)$ can be defined as a function that maps a state s to a particular value that depicts the expected return starting from state s and following policy π. This can be expressed as $v_\pi(s) = \mathbb{E}_\pi\{G_t|S_t = s\}$ The action-value function $q_\pi(s)$ is defined as a function that maps a state-action pair (s, a) to the expected return if the agent starts from state s, performs action a, following a given policy π. This can also be expressed as $q_\pi(s) = \mathbb{E}_\pi\{G_t|S_t = s, A_t = a\}$ In order to update the value function Monte-Carlo update [96] and the temporal-difference (TD) update [90] can be used.

3.3.3. Policy search and optimization

The policy search and value function approximation are two important features in RL. A mixture of both approaches is common in the literature as well. In essence, RL aims to learn a policy, $\pi(a|s)$, that describes the distribution through state-dependent actions or learns the vector of parameters θ of the function approximation. By finding an optimal policy, the agent can determine the best action for each state to maximize the rewards. The policy $\pi(a|s,\theta) = \Pr\{A_t = a/S_t = s, \theta_t = \theta\}$, probability of performing action a when in state s and parameters θ . Policy search focuses on finding good parameters for a particular policy parameterization such as the neural network weights. Optimal policy search can be accomplished through gradient based methods using back propagation or gradient free methods. According to Ref. [35], the policy-based method directly optimizes the policy $\pi(a|s;\theta)$ and updates the parameters θ by maximizing the loss function . REINFORCE in Ref. [35] is an example of a policy gradient method. Other policy gradient algorithms include Vanilla policy Gradient, TRPO, PPO. Value-based methods include TD learning [90] and Q-learning [86]. Please see the work in Refs. [35,94,101,102] for details.

3.3.4. The exploration vs exploitation dilemma

To maximize rewards, the RL agent needs to balance between trying new things with sub-optimal actions i.e., exploration and gradually favour those actions that seem to be the best or prefer actions that the agent already knows from experience that they return high rewards i.e., exploitation. Sub-optimal actions could lead to lower rewards in the immediate future but generate a good strategy that empowers an improved policy in the long term. The multi-arm bandit problem [98] and finite state space MDPs [99] have often been used to investigate the balance between exploration and exploitation. Simple methods such as Softmax and $\varepsilon-$ greedy with a fixed probability $0 < \varepsilon > 1$ [100] are used to examine the exploration vs exploitation dilemma. 10% of the actions will be exploration, i.e., taking random actions and 90% of actions will be exploitation, i.e., choosing the action that returns the best long term

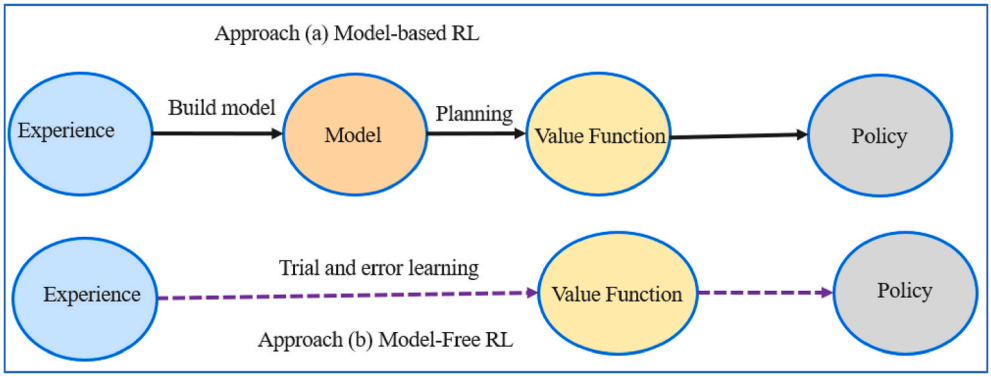


Fig. 6. Model - based RL vs Model - free RL.

reward if ε is set to 0.1. A similar method is $\varepsilon-$ greedy with decay where the ε is set to 1 and gradually decreases with each training episode when it's multiplied with *decay parameter* < 1. Exploration is likely to happen at the beginning as the agent learns about the environment, whose probability will dimmish and eventually taking only the most profitable actions. The Softmax method uses action-selection probabilities that are determined by classifying the value function estimates using a Boltzmann distribution [35]. Sutton and Barto's Book in Ref. [35] provides more details on the exploration vs exploitation phenomenon.

3.3.5. Off-policy RL vs on-policy RL vs offline RL

Policy function maps states to actions. In off-policy RL, the algorithm evaluates and improves the update policy which is different from the behaviour policy. The advantage of this separation is that the update policy can be deterministic (e.g., greedy), while the behaviour policy scans for all the possible actions [35]. The experiences $\ll s, a, r, s^{'} \gg$ obtained through the agent's interaction with the environment using the behaviour policy are stored in replay buffer to update the target policy and the agent's subsequent interaction with the environment using this new policy. DQN is an example of an off-policy algorithm that updates the policy or its Q function by training the replay buffer. Q-learning is also off-policy as it updates its Q-values using the Q-value of $s^{'}$ and the greedy action $a^{'}$ using:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(R + \gamma max_{a'} Q(s^{'}, a^{'}) - Q(s,a)).$$

In on-policy RL, the update policy and the behaviour policy are the same. When the agent interacts with the environment, it collects samples, that are used to improve the same policy the agent is using for selecting actions then determines what to do next. SARSA [92] is an example of on-policy algorithm. It updates the Q-function with actions of the same policy as $Q(s,a) \leftarrow Q(s,a) + \alpha(R(s,a) + \gamma Q(s^{'}, a^{'}) - Q(s,a))$ where $a^{'}$ and a need to be chosen according to the same policy. Other examples of on-policy algorithms include policy iteration, PPO, TRPO etc.

In offline RL, commonly known as batch RL, the agent uses previously collected data, without additional online data collection. The agent can't interact with the environment and can't use the behaviour policy to collect additional transition data. The learning algorithm is provided with a static dataset of fixed interactions with the environment and learns the best policy using this dataset. This method is like a supervised learning phenomenon.

3.3.6. Model-based vs model-free RL approaches

Model-based and model-free algorithms are key critical components of the RL framework. In the model-based RL approach, the model of the environment defined by the state space S, the action space A, the transition matrix T, and the reward function R, is formed from the experience of the agent. This model is used to estimate the value function. Policy iteration and Value iteration are typical examples. All these algorithms take advantage of the model's next state and reward prediction or distribution to calculate the optimal action before performing it. Model-based approaches exhibit the following advantages in real world

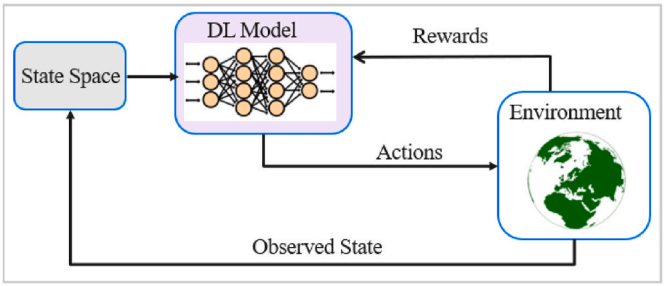


Fig. 7. Deep Reinforcement learning process.

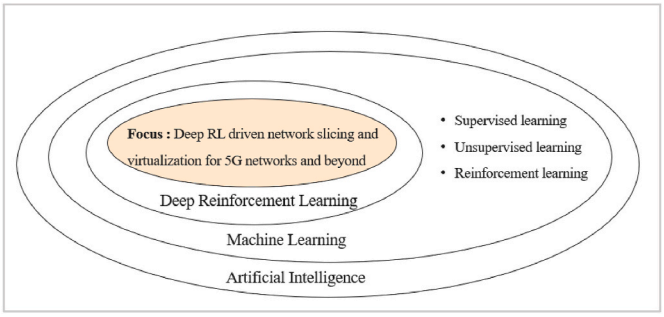


Fig. 8. Relationship among deep RL, DL, RL, SL, USL, ML and AI

sequential decision making problems (a) yield good results, (b) optimized utilization of data [29], (c) leverage domain knowledge programming to accelerate learning [95], (d) the learned model may assist the system when the objectives change [95]. On the other hand, model-based RL methods face the following limitations. For example, (a) an accurate model of the underlying system is required to improve learning, which may not be the case or may take a long time to acquire, such may be the case in dynamic network environments such as 5G (b) it's often very complicated to resolve non-convex optimization problems. Offline calculations could help address this challenge [94], (c) learning from experience is challenging, learning rules will need to re-coded to reinstate optimal behaviour if the model is incorrectly optimized [95].

In model-free RL, the transition matrix T and reward $\mathscr{R}$ are unknown to the agent. It goes directly from experience to estimating a value function or an optimal policy by interacting with the environment. In essence, the set of trajectories collected from the environment provides the experience data the agent needs to enhance learning. In the Q-learning example, the agent estimates the Q-value or the approximate value function for each (state, action) pair and provides the optimal policy by selecting the action that gives the highest Q-value given the state of the agent. Contrary to model-based algorithms, Q-learning cannot predict the next state and value before taking the action. Consequently, model-free techniques require a vast wealth of experience. Leveraging a combination of model-based and model-free algorithms strengths counterbalances their mutual weaknesses. Please see details in Ref. [35]. The Model-free approach requires a complete exploration of the environment. For this reason, when applied to complex systems like 5G networks, the learning phase can be very inefficient, requiring a considerable amount of time before convergence. The novel ML techniques such as Deep Q-learning discussed in section 3.4, that approximate the Q-values with a DNN, can overcome this issue, and enable a complete exploration thus minimizing the approximation loss of the DNN. A summary of the model-based RL and model-free RL approaches is shown in Fig. 6.

3.3.7. Q-learning overview

Watkins doctoral thesis [82] pioneered the Q-learning algorithm. Q-learning belongs to model-free, temporal difference (TD) update, off-policy stochastic descent RL algorithms and consists of three major steps.

- The agent chooses an action a_t in state s_t according to some policy like ε -greedy.
- The agent obtains a reward $R(s,a)$ from the environment, and the state transitions to the next state $s^{'}$.
- The agent updates the Q-value function in a TD manner as follows:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(R(s,a) + \gamma \max a^{'} Q(s^{'}, a^{'}) - Q(s,a))$$

Q-learning may also be thought of as an asynchronous dynamic programming technique (DP). It allows agents to learn how to act

optimally in Markovian domains by experiencing the consequences of their actions rather of having to develop maps of the domains [103]. It makes no assumptions about the agent's knowledge of the state-transition and reward models. However, the agent will learn what acts are good and harmful via trial and error [35]. With this approach, a look-up table (Q-table) comprising of state-action pairs and potential rewards is created for the RL algorithm to learn from. The Q-learning algorithm suffers from the "curse of dimensionality" in large state and action spaces, as traversing all the state-action spaces is not only memory intensive but also slows down convergence. This makes Q-learning only suitable for small scale networks. Researchers have advocated for the function approximation approach to address this limitation. Functional approximation reduces the number of unknown parameters to a vector with dimension n and the related gradient method further solves the parameter approximation in a computationally efficient manner. The non-linear function approximation methods that use deep neural networks (DNN) can efficiently be used for value approximation in the RL framework.

3.4. Deep RL fundamental principles

Deep RL evolved from the use of DNNs (non-linear method) to approximate either the (a) value function, $v\hat{}(s;\theta) or q\hat{}(s,a;\theta)$, that shows how good the states or actions are or (b) the policy $\pi(a|s;\theta)$, which describes the agent's conduct or (c) the model (state transition function and reward function) of the given environment. Deep RL uses deep learning (DL) tools to extract features from complex high dimensional data and transforms them to a low-dimensional feature space, and then uses RL to make the decisions. Please see Fig. 7 for a summary of the Deep RL process.

DNNs help the agent to extract the most relevant features from the state representation. Here, the parameters θ are the weights in the DNN. Stochastic gradient descent is utilized to update weight parameters in Deep RL. As discussed in Ref. [137], typical examples of DNN include Convolution neural networks (CNN), Recurrent Neural networks (RNN) and many others. The deadly triad of function approximation, bootstrapping and off policy learning to address TD learning failures in certain use cases is documented in Ref. [35]. Recent studies such as the Deep Q-Network [105], AlphaGo [17–19] and others have improved the learning and achieved tremendous results. The work in Ref. [105] introduced the Deep Q-Network (DQN) and pioneered research in the Deep RL field. The DQN applied DL to Q-learning, by approximating the Q-table using a deep neural network.

In the areas of experience replay [106] and network cloning [108], the DNN has achieved significant development that make off-policy Q-learning attractive. The Deep Q-Learning (DQL) agent gathers experience data (state-action-reward values) and trains its policy in the background. In addition, the learnt policy is saved in the neural network (NN) and may be easily transferred across instances. In the network slicing applications, the DQL agent may function effectively and make resource allocation choices in a timely manner based on its already learnt policy. That way, complex slice orchestration and resource allocation challenges in 5G and beyond network slicing might benefit from such a strategy. With its capabilities to handle large state-action spaces, Deep RL is a candidate for use in heterogenous, dynamic 5G and beyond network slicing, the core of this study. Deep RL's relationship with ML is summarized in Fig. 8.

Table 3
Overview of Deep RL Applications in 5G Cellular networks.

Author	Objective/goal	Approach
Zhao et al. [61]	Maximize network utility while satisfying QoS	DDQN
Zhang et al. [66]	Real time channel information acquisition, User association	Deep RL
Yu et al. [85]	Resource management and network optimization of 5G HetNets	Deep RL
Mismar et al. [97]	Beamforming, power control, and interference coordination to maximize SINR	Deep RL

Various enhancement to the DQN such as the double DQN (D-DQN) [109] have emerged to improve performance and simulation results on Atari games have shown better performance when compared with the DQN. The authors in Ref. [115] attempted to understand the success of DQN and reproduced results with shallow RL. The authors in Ref. [116] proposed a hybrid combination of the policy gradient and Q-learning called PGQ method to improve the performance of the policy gradient. The PGQ performed better than actor-critic (A3C) and Q-learning on Atari Games. The authors in Ref. [118] designed a better exploration strategy to improve the DQN.

3.5. RL and deep RL applications

Deep RL techniques are being applied to various fields such as economics [31], finance [120], industrial automation [121], robotics [105], gaming [17–19], green communications [25], computer vision [122], natural language processing (NLP) [123,124] and many other domains. An overview of practical applications of Deep RL can be found in Ref. [125] for further reference.

3.6. RL and deep RL research assistance

Recent breakthroughs in Deep RL have created a lot of euphoria in RL and Deep RL research. Several platforms are available to develop and train RL and Deep RL agents. An outline of some key platforms is provided for interested readers and researchers.

- The OpenAI Gym [127,128] is an opensource toolkit for RL environments that can be used for the development, comparison, testing and reproduction of RL algorithms.
- The Arcade Learning Environment (ALE) [126] is a framework that comprises of Atari 2600 game environments to support the development and evaluation of AI agents. It's a good testbed for solutions to RL, imitation learning, and transfer learning problems.
- MuJoCo [129], acquired by DeepMind in 2021, is a physics engine that can be used in the research and development of RL and Deep RL solutions in robotics, biomechanics, graphics, animation, etc. Full open sourcing will be available in 2022.
- DeepMind Lab [130] can be used to develop and test 3D gaming agents.
- DeepMind Control Suite [131], developed to serve as performance benchmarks for RL agents, can simulate RL environments using the MuJoCo physics engine.
- Dopamine [104] is TensorFlow-based RL framework developed by Google for testing RL algorithms. It supports C51, DQN, IQN, Quantile (JAX) and Rainbow agents.
- Extensive, Lightweight and Flexible (ELF) [132] platform is a pytorch library for game research. ELF OpenGo [133] is a reimplementation of AlphaGo Zero/Alpha Zero using the ELF framework.
- Ns3 Gym [11] is compatible with Python, and C++ languages and combines NS3 and OpenAI Gym to test RL agents for solving network communication problems.
- AirSim [15] is an opensource platform driven by RL, Deep RL, and computer vision, for developing algorithms used for autonomous vehicles such as drones, cars, etc. It's compatible with C++, C#, Python, Java and flight controllers such as PX4.
- Reco Gym [16], based on the OpenAI Gym, is a platform for developing recommender systems using traffic patterns for mobile advertising and e-commerce. It uses the multi-bandit problems to accomplish this purpose and supports the python language.
- Other platforms of interest include Vizdoom [20] for RL agents to play the Doom game, Deepmind OpenSpiel [21] for python based RL development, Facebook's ReAgent [23] for E2E large scale RL

Table 4
Useful materials and resources in AI/ML/DL/RL/Deep RL.

Authors	Main Domain	Material
Russel and Norvig's	Foundations, principles, and theories of AI	[73]
Jordan, and Mitchell's paper	Machine learning paper	[74]
LeCun et al. famous paper	Deep learning paper	[134]
Sutton and Barto RL book	Fundamentals and recent advances in RL progress, e.g., in deep Q-network, AlphaGo, policy gradient methods etc.	[35]
Bishop (2006)	Pattern recognition Machine learning textbook	[135]
Littman (2015)	Machine learning	[136]
Goodfellow et al. (2016)	Deep learning book	[137]
Sergey Levine's (2018)	Deep RL Course	[138]
Yuxiliu (2018)	Deep RL Tutorial	[139]
David Silver (2015)	RL course	[140]
Neeraj et al. (2020)	Scientometric Assessment of Global publications output between 2009 and 2018	[141]
Hastie et al. (2009),	Machine learning textbook	[142]
Murphy (2012)	Machine learning textbook	[143]
James et al. (2013)	Introduction to ML	[144]
Domingos (2012)	Practical machine learning	[145]
Zinkevich (2017)	Practical machine learning in Tensor flow	[146]
Ng (2018)	Practical machine learning.	[147]
Schulman (2017)	Practical deep learning and deep RL	[148]
Lilicrap (2015)	Deep RL paper	[149]

production, Stanford's OpenSIM [24] for RL powered locomotion tasks and Microsoft initiated Project Malmo [26] for RL powered complex structures with intricate features.

3.7. Considerations for deep RL in 5G mobile communication

5G and beyond networks need to support different applications and use cases with different RAN and QoS requirements. Due to the complexity of these networks, manual network operation and maintenance is considered difficult, ineffective, often suboptimal, and costly. Deep RL, a combination of RL and DNN, provides an autonomous solution to mobile network operational challenges. With the ability to operate in higher-order spaces, Deep RL enhances the robustness and effectiveness of 5G and beyond mobile networks in a variety of scenarios. For example, in Ref. [61], Multiagent RL, which uses the DDQN approach, maximizes network utilities while preserving QoS for user devices on heterogeneous networks.

The author of [66] uses two Deep RL algorithms that use historical data to make appropriate decisions to obtain real-time channel information for user associations in a symbiotic wireless network. Deep RL can also address challenges in the areas of resource management and network optimization, as described in Ref. [85]. In the absence of a prior traffic models, Deep RL can be used to address the challenges of optimizing complex non-convex and convex network problems such as user association, interference management, power control and many others. For example, in Ref. [97], the authors formulated a joint beamforming, power control, and interference coordination problem as non-convex optimization problem for maximizing signal to noise interference radio (SINR), and solved the problem using Deep RL. Table 3 provides a summary of the reviewed Deep RL usage in 5G cellular networks.

3.8. Recent innovations in RL and deep RL

The RL framework has recently become one of the most exciting areas of ML research. With the ability to learn like a human through trial and error rather than relying on datasets, the RL framework is a promising contribution to achieving the AI vision and is a powerful tool for automating the life cycle of 5G network slices. Consequently, large technology companies such as Facebook, Google, DeepMind, Amazon, Microsoft, as well the academia and industry are heavily investing in RL research.Recent innovations in Deep RL are discussed below:

3.8.1. Transfer learning to accelerate deep RL in 5G network slicing

The main goal of transfer learning [13] is to improve the convergence speed of the model by reusing previously acquired knowledge to solve similar problems rather than starting from scratch. For example, by reusing the policy learned by a DRL agent in the expert base station to facilitate training of the newly deployed DRL agents in the target learner

Table 6
A high level summary of features and network evolution up to 5G.

Features	2G	3G	4G	5G
Rollout	1993	2001	2009	2018
Technology	GSM	WCDMA	LTE, Wimax	MIMO, mmWave
Access system	TDMA, CDMA	CDMA	CDMA	OFDMA, BDMA
Switching type	CS for voice and PS for data	Packet switching except air interface	Packet switching	Packet switching
Internet service	Narrowband	Broadband	Ultra-Broadband	Wireless broadband
Bandwidth	25 MHz	25 MHz	100MHz	30 GHz–400 GHz
Benefits	Multimedia internet access and SIM	High security, internal roaming	High speeds, High speed handoffs, IoT	Extremely high speeds, low latency, massive device connectivity
Applications	Voice calls, SMS	Video conferencing, mobile TV, GPS	High speed applications, mobile TV, wearable devices	High resolution video, streaming, remote control of vehicles, robotics, e-Health

Table 5
Summary of the characteristics of common RL/Deep RL algorithms.

Algorithm	Description	Policy	Action space	State space	Operator
Q-Learning [82]	Quality learning	Off-policy	Discrete	Discrete	Q-value
SARSA [92]	State-Action-Reward-State-Action	On-policy	Discrete	Discrete	Q-value
Monte Carlo [96]	Monte Carlo method	Either	Discrete	Discrete	Sample averages
DQN [105]	Deep Q Network	Off-policy	Discrete	Continuous	Q-value
DDQN [109]	Double Deep Q Network	Off-policy	Discrete	Continuous	Q-value
DDPG [164]	Deep Deterministic Policy Gradient	Off-policy	Continuous	Continuous	Q-value
A3C [157]	Asynchronous Advantage Actor - Critic Algorithm	On-policy	Continuous	Continuous	Advantage
A2C [157]	Advantage Actor - Critic Algorithm	On-policy	Continuous	Continuous	Advantage
TRPO [172]	Trust Region Policy optimization	On-policy	Continuous	Continuous	Advantage
PPO [171]	Proximal Policy optimization	On-policy	Continuous	Continuous	Advantage

base station, the authors in Ref. [114] show that transfer learning optimizes Deep RL driven resource allocation in 5G network slicing scenarios.

3.8.2. Sequential learning made easier

To simplify the implementation of complex sequential learning models such as model-based RLs, batch RLs, hierarchical RLs, and multi-agent RL algorithms, Meta recently released the SaLinA [173] library, an extension of PyTorch that can run on multiple CPUs and GPUs. IBM recently published TextWorld Commonsense (TWC) in Ref. [174]. This is a gaming environment that introduces common sense to RL agents. This method trains and evaluates RL agents with specific common sense about objects, their attributes, and offerings. With the introduction of several RL baseline agents, IBM provides a new platform for addressing a variety of decision-making challenges.

3.8.3. Innovations in self supervised learning

Self-supervised learning aims to find features without the use of manual annotations [175,176] and recent studies have shown that self-supervised features that are competitive with supervised features can be learned. This field has thus been recently complimented with some interesting innovations. For example, the authors in Ref. [177] introduce reversible-aware RL. Their research investigates the use of irreversibility to drive sequential decision-making. By using reversibility in action selection, the search for unwanted irreversible actions can be enhanced. The game of Sokoban puzzles could be one area where their work may be useful.

3.8.4. Multitasking robotic RL

To simplify robotic training, the authors in Refs. [215–218] address this issue in their research on MTOpt and Actionable model solutions. The former is a multitasking RL system for automatic data collection and multitasking RL training, and the latter is a data collection mechanism for episodes of various tasks with a real robot and demonstrate multi-tasking RL in action. It helps the robot learn new tasks faster. DeepMind recently published RGB-Stacking, a benchmark for visual-based robotic operation in Ref. [225]. DeepMind uses RL to train robotic arms to balance and stack objects of various geometric shapes. The variety of objects used, and the number of empirical evaluations performed made this RL-based project unique. The learning is divided into three phases, simulation based training using standard RL algorithms, training new policy in simulation using only realistic observations, and finally using this policy in real robots to gather data and train an improved policy based on this data. More details on these new innovations can be found in Refs. [215–225]. A phase by phase approach allowed DeeepMind to break the task into subtasks and resolve the problem in a quicker and more efficient manner.

3.8.5. Deep RL innovations in gaming

Even with the success in the Go games, DeepMind couldn't train an agent to learn new games without repeating the RL process from the beginning. DeepMind's recent innovation has allowed agents to be trained to play Atari games, adapt to new conditions that flexibly adapt to new environments. Deep RL was central to this research and played an important role in training agent's neural networks. The evolving RL algorithm, recently investigated in Ref. [227], showed the gaming research community how to use graph representations and apply AutoML community optimization techniques to learn analytically interpretable RL algorithms. The population of the new RL algorithm is developed using regularized evolution [226]. The mutator modifies the best performing algorithm to create a new one. The performance of the algorithms is evaluated over various training environments and the population is updated. Their method reuses existing knowledge by updating the population with a known RL algorithm rather than starting from scratch. This method can be used to improve the RL algorithms in complex environments with visual observations such as Atari games.

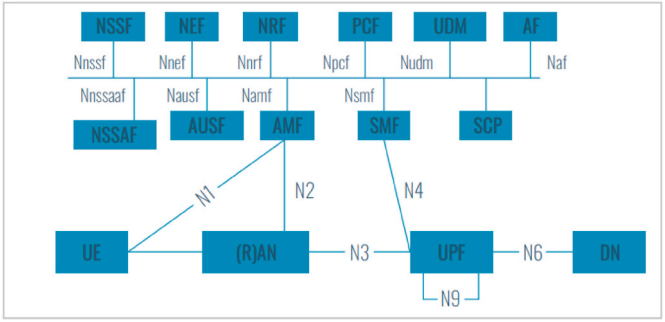


Fig. 9. 5G service based architecture [221].

3.9. Useful study, reference materials and resources

RL and Deep RL research efforts have led to a multitude of publications. Some useful AI, ML, DL, and Deep RL books, papers, courses are presented in Table 4 for the reader's reference.

3.10. Summary and lessons learned

The objective of artificial intelligence is to develop intelligent and helpful agents that can solve real-world problems. An RL agent performs a series of activities and monitors states and rewards using the basic components of a value function, policy, and a model. A prediction, control, or planning problem can be written as an RL problem using model-based or model-free methods. A fundamental dilemma between exploration and exploitation exists in RL. DNNs improve the learning process in RL and help address the "curse of dimensionality," in high order spaces. As the RL field continues to develop, several modern RL models are (a) model-free, (b) can be applied in various dynamic environments, and are (c) capable of responding to new and unseen states. The combination of these RL capabilities with DNNs have made Deep RL's application in the management and orchestration of 5G network slicespromising. A summary of the characteristics of some common RL/Deep RL algorithms is presented in Table 5.

4. The 5G cellular network

This section provides an overview of 5G, elucidates on the softwarization and programmability principles critical to realizing effective 5G network slicing and virtualization.

4.1. Historical evolution of cellular networks towards 5G

The history of wireless communication dates to Marconi, an Italian engineer, who first transmitted a wireless message over an electromagnetic wave in 1898. Since then, wireless communication has transformed society and started a digital revolution that has impacted human civilization for decades. The first generation (1G) started the wireless evolution and as times went by the data rates, mobility, coverage, latency, and spectral efficiency improved till the fifth generation (5G) of wireless networks.

The SDOs such as 3GPP [69], ITU-T [1] and many others have ensured that a new generation of wireless technologies is standardized every 10 years. Licensed, and unlicensed spectrum is at the centre of these evolutions. Following the successful 5G standardization, rollout and monetization, studies on sixth generation (6G) of networks has commenced. 5G transcends earlier generations and represents a paradigm shift in mobile network communication. A high level summary of some of the key features of mobile network generations up to 5G is given in Table 6.

4.2. The 5G service based architecture (SBA)

The 5G System architecture is intended to offer data connectivity and services while also allowing deployments to adopt network function virtualization (NFV) and software defined networking (SDN). These technologies separate the data plane from the signalling control plane, providing independent scalability and flexible deployments such as centralized or distributed deployments, paving the way for the implementation of various 5G services. Network functions (NF) and service-based interfaces make up the architecture.

We present the 5G SBA network functions in Fig. 9 as presented in Ref. [221]. A brief introduction to the NF pertinent to this study will follow.

4.2.1. The 5G core network (5GC)

The 5G SBA architecture consists of NFs interconnected via standard interfaces. The key NFs pertinent to our study include the following: (a) Access and Mobility Management Function (AMF): Responsible for security and idle mobility in the non-access layer (NAS), (b) User Plane Function (UPF): manages all the data packets traversing the RAN to the internet in terms of QoS, inspection, forwarding and usage reporting, (c) Session Management Function (SMF): This network function is responsible for the allocation of an IP address to a user equipment (UE) and controlling the PDU session, (d) Network slice selection function (NSSF): Used by the AMF to select network slice instances that serve a specific UE. The NSSF determines the Network Slice Selection Assistance Information (NSSAI) that can be delivered to the device or UE. If the current AMF can't support all network slice instances for a particular device, the NSSF can be used to assign the appropriate AMF.

We recommend the 3GPP TS 23.501 v16.5.1 and 3GPP TS 23.501 v16.5.1 [221] documents for more detailed descriptions on all the NF and interfaces.

4.2.2. The 5G radio access network (5G NG-RAN or 5G NR)

The next-generation NodeB (gNB) and the next generation evolved NodeB (eNB) are responsible for the DL/UL data transmission between the 5GC and the UE in a non-standalone 5G architecture (5G NSA). They also perform radio resource management in the 5G network. The gNB and eNB are designed based on 5G New Radio and long term evolution (LTE) protocols respectively. The 5G NG-RAN is designed with a flexible Radio Resource structure based on scalable orthogonal frequency division multiplexing (OFDM) numerologies. Each of the resource blocks (RB) consists of 12 consecutive frequency subcarriers with scalable subcarrier spacing. The sub carrier spacing supported for each RB includes 15 kHz, 30 kHz, 60 kHz, 120 kHz, and 240 kHz. 5G supports scalable transmission timeslot sizes for RBs including 1 ms, 0.5 ms, 0.25 ms, 0.125 ms and 0.0625 ms. The various options introduced in the OFDMA numerologies will enable 5G to support a plethora of services with diverse network requirements. For example, RBs with a large subcarrier spacing and short timeslot size are suitable for high-bandwidth, low-latency applications such as video games, whereas RBs with a short subcarrier spacing and long-time timeslot size are suitable for short-packet, less delay sensitive applications such as IoT use cases.

4.3. The need for AI/ML introduction in the 5G architecture

The 5G system is designed to provide huge throughputs, support diversified enterprise services, and connect a magnitude of user equipment (UE) and devices. The enormous increase in data traffic and use cases means increased complexity in network management and monitoring. The AI/ML techniques are envisioned to play a critical role in solving previously considered NP-hard, and complex network management and optimization problems. Consequently, the ITU Y.3172 standard provides guidelines on the use of ML in next generation networks. ML allows the systematic extraction of useful information from traffic data and the automatic discovery of patterns and connections that would otherwise be impossible if done manually. By supporting analytics and knowledge discovery, ML tools will provide network operators with more accurate information to quickly make decisions and effectively monetize 5G and beyond networks. The ITU-T steering committee in Ref. [113], established in 2018, is a focus group that advocates for ML solutions in 5G and beyond networks. The preliminary studies on 6G in Refs. [158,159,233], have all shown that AI/ML techniques will play a pivotal role in the next generation networks.

5. Network slicing principles in 5G and beyond

This section elucidates on the topic of 5G network slicing and virtualization, diving into its history, principles, key technology enablers and paves the way for further discussions on the core topics of this paper.

5.1. Network slicing historical evolution

In the 1960s, IBM's CP-40 operating system supported time sharing, virtual storage and simultaneous user access [160]. This was the first form of network sharing. Planetlab [245] pioneered slicing in the computer architectures in 2003. In their work slices were separated and tailored to a particular application. This laid the foundation for the current NGMN slicing concept. 2008 saw the acceleration of research

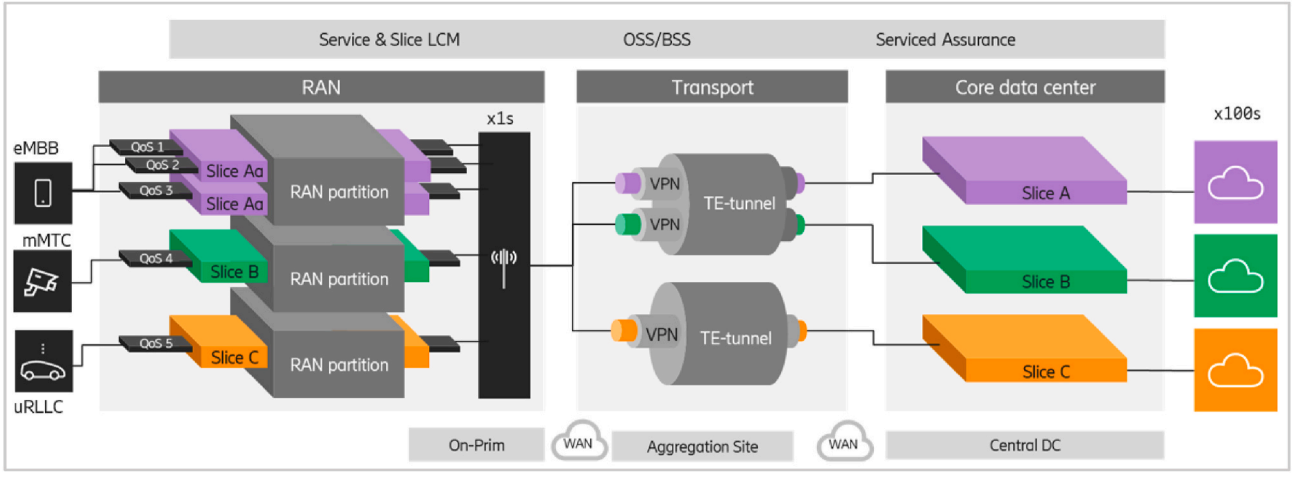


Fig. 10. The E2E network slicing with each slice representing a particular use case under the enhancement mobile broadband (eMBB), massive Machine type communication (mMTC) and ultra-reliable low latency categories.

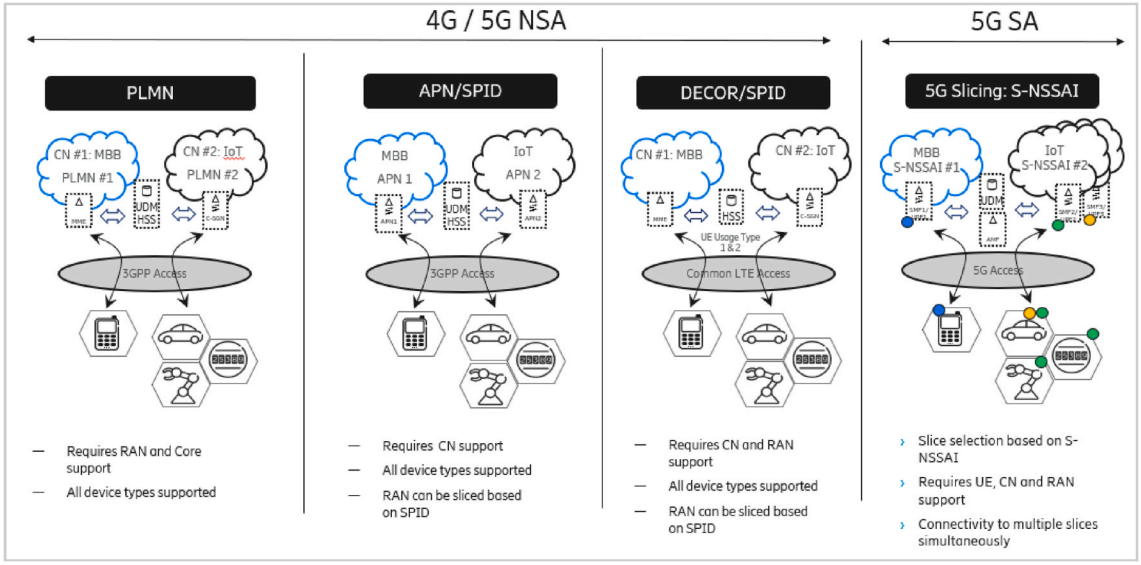


Fig. 11. Summary of network slicing mechanism.

into virtualization technologies mainly spearheaded by the Global Environment for Network innovations (GENI) project [163]. The introduction of SDN in 2009 accelerated the development of network programmability and software that underpins scalable and customizable network segments. Equipped with the fundamental concepts above, several SDOs including ETSI (European Telecommunications Standards Institute) [261], IETF (Internet Engineering Task Force) [210], 3GPP (3rd Generation Partnership Project) [212], NGMN (Next Generation Mobile Networks) [10] and ITU-T (ITU Telecommunication Standardization Sector) [161], attempted to define the slice concept in the mobile networking domain. The single access point name (APN) used by mobile consumers to share various voice and data services was the first network slicing mechanism in the 3GPP domain. Multi-Operator Core Networks (MOCN, 3GPP TS 23.251) which was later introduced in the 2010 and beyond by 3GPP allowed operators to share RAN resources while maintaining their core networks. Dedicated Core Network (DECOR) and enhanced DECOR (eDECOR) supported core network slicing as early as LTE in 3GPP release 13 and 14 respectively. The first version of 5G network slicing, a product of the NGMN [10] project, was standardized in 3GPP release 15 in 2018. Their new addition to the existing slicing mechanisms was the Single Network Slice Selection Assistance Information (S-NSSAI). This paved the way for the establishment of logical/virtual enterprise networks on 5G physical infrastructure. Preliminary studies on 6G in Refs. [158,159,233], have all shown that network slicing will continue to play a critical role in the beyond 5G networks.

5.2. Principal concepts

Coined and first introduced by the NGMN project [10], network slicing aims to divide the physical network infrastructure into multiple virtual networks consisting of common E2E RAN, core network (CN) and transport resources to support diverse business services, enterprise applications and use cases. The network slices are mutually isolated, elastic, with independent control and management and can be created on demand [162]. A high level illustration of an E2E network slice is shown in Fig. 10. Several SDOs including ITU-T [1], the 3GPP [212], ETSI [261], IETF (Internet engineering task force), 5GPP (Fifth generation Partnership project) invested resources in developing network slicing foundations, principles, and architecture. For example, the NGMN network slicing architecture consists of (a) service layer that represents the requirements that network operators, or a third party must meet, (b) network slice instance layer created based on a network slice blueprint to describe the requirements of network slice instances and iii) network resources layer which handles multidimensional resources that could be allocated to the network slice. These include network, computing, and storage resources. NFV is used to create virtual network functions (VNFs) that meet the needs of network slice providers and deploy those network functions (NFs) to cloud networks to reduce TCO. A core network slice basically forms a virtual network consisting of

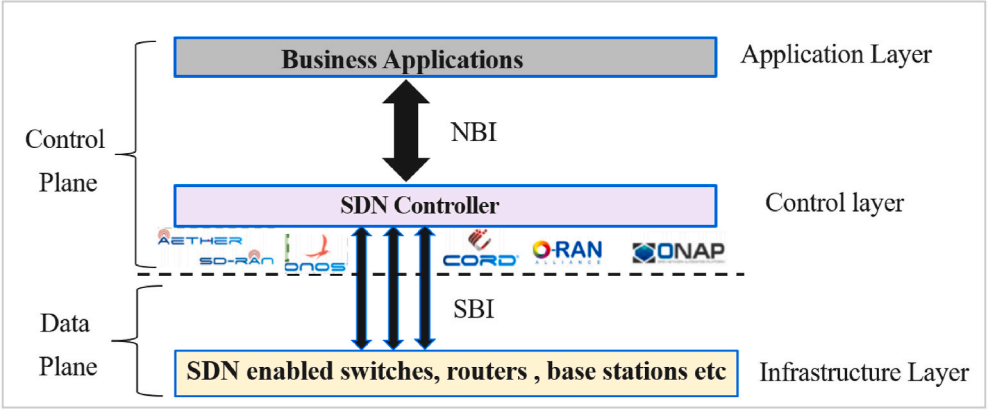


Fig. 12. Software defined networking diagrammatic illustration.

one or more ordered VNF service chains. RAN slicing involves the abstraction and sharing of RAN nodes, sharing radio spectrum resources in a dynamic logical manner. Various services classified under eMBB, mMTC, uRLLC categories can be instantiated on the slices per user, per device or even per application [165]. Consequently, scalable, and flexible network slices will be required to manage resources needed at peak traffic demand. The resources can be adjusted or removed by management functions. The slices will enable service differentiation based on predefined SLA. Partitioning the physical infrastructure into multiple logical E2E networks that serve divergent enterprise use cases, championed the concept of Network slice as a service (NSaaS), enabling mobile network operators to increase revenue and profits in 5G and beyond networks. The sharing of infrastructure helps InPs to reduce network TCO [62].

5.3. Network slicing mechanism per domain

3GPP defines a network slicing framework for the 5G standalone architecture (5G SA) based on the S-NSSAI parameter to enable the creation and management of network slices via RAN and the core networks (CN). The CN and RAN can also be sliced based on the Public Land Mobile Network (PLMN-ID) identifier. While subscriber profile identifier (SPID)-based slicing only partitions the RAN, DECOR partitions the core network. In the 5G non standalone architecture (NSA) architecture, CN slice selection using PLMN-ID is controlled by the eNodeB as signalling is transported via the Long Term Evolution (LTE) leg. Using DECOR and eDECOR, the UE can only connect to one network slice at a time. After the introduction of the SNSSAI mechanism, 5G capable UEs can now simultaneously connect to multiple slices, as defined in the 3GPP specifications. Different partitioning mechanisms have different features such as isolation, data protection, concurrent connections, and UE support. For example, APN differentiation typically provides partitioning of the packet core gateway (PGW) resources, while DECOR and eDECOR enable selection of dedicated CN resources such mobility management entity and gateway. The PLMN-ID can be used to completely isolate the resources used for the dedicated network. A summary of 5G RAN and CN slicing mechanisms is provided in Fig. 11.

In summary, the 5G slice selection mechanism, standardized in 3GPP, is characterized by:

- Enabling simultaneous UE connections to multiple network slices.
- Ability to provide different isolation levels between network slices.
- RAN network slicing (slice-aware resource allocation and AMF routing) using a common parameter for the RAN and core network.
- Allows different slices to handle simultaneous PDU sessions using the same or different Data Network Names (DNN). PDU sessions cannot be shared by different network slice instances.
- Serving different slices to the same UE. Slices are selected to better suit the needs of the UE connectivity service at each point in time.
- Network slice selection assistance information (NSSAI) as defined in 3GPP TS 23.501. It consists of a list of S-NSSAI and supports slice selection. 3GPP has defined commonly used slice service types (SSTs) such as eMBB and mMTC, which are components of S-NSSAI. S-NSSAI can also include a differentiator (SD) to allow multiple network slices to be created with the same SST.
- Support many-to-many cardinality between SNSSAI and Network Slice Instances (NSIs) deployed to serve that S-NSSAI, enabling UE simultaneous access to multiple slice instances.

5.4. Technology enablers for network slicing

This section provides an overview of network slicing and virtualization technology enablers.

5.4.1. SDN and NFV

There is now consensus in the SDOs and industry that softwarization, driven by SDN and NFV technologies, will continue to be an important enabler in slicing 5G networks and beyond. Both SDN and NFV provide programmable control and resource management functionality that are essential for dynamically implementing 5G network slices. When SDN and NFV are combined with various cloud computing delivery models such as IaaS [178], PaaS [178] and SaaS [178], more sophisticated infrastructures and solutions can be implemented. As illustrated in Fig. 12, the SDN is used to separate the network control from the data control plane. It also separates the routing and control procedures from hardware-based forwarding operations. Consequently, softwarized network control procedures can be implemented with an abstracted view of the underlying physical infrastructure. The centralized SDN network can make intelligent, and automated network management, operation, and optimization decisions from a global network perspective. 5G also adopts a similar dual plane separation mechanism to split the RAN and mobile edge hardware from the network and service functions. Whereas the slice orchestration capabilities are beyond its capabilities, SDN enables the implementation of 5G network slicing

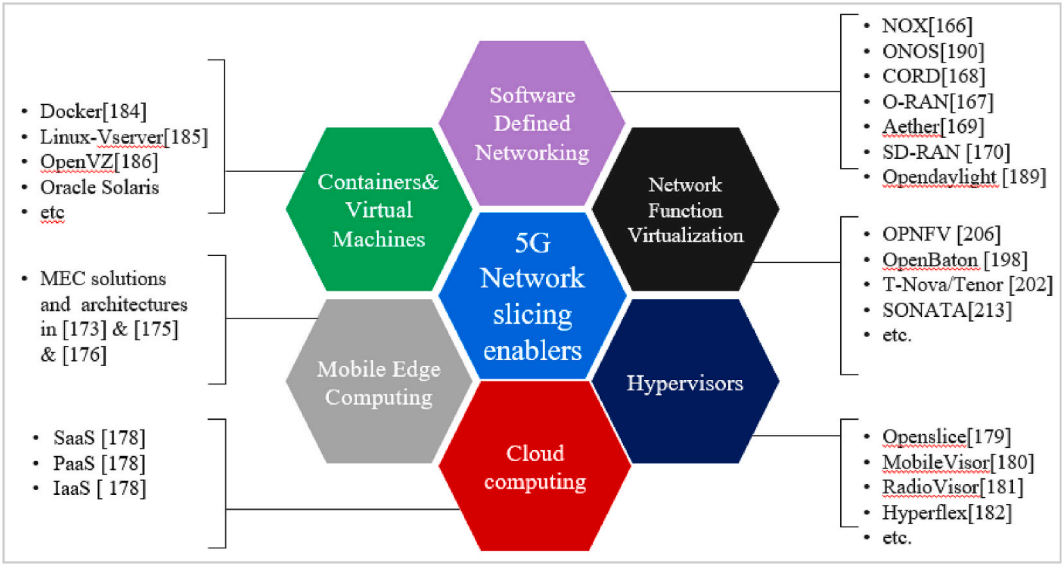


Fig. 13. Summary of 5G network slicing enablers.

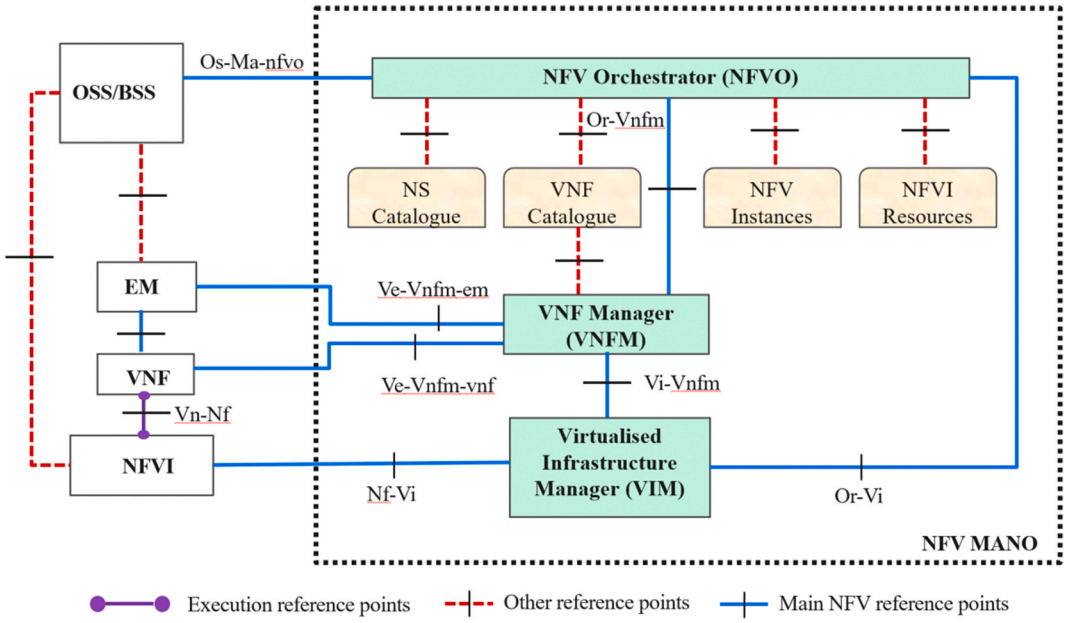


Fig. 14. The ETSI NFV MANO system architecture as proposed in [187].

procedures. It supports the flexibility and programmability to simplify network management. NOX [166] is an example of the first SDN controller based on the OpenFlow protocol. Since then, the industry has developed several platforms, primarily for cellular networks, including the Open Networking Operating Systems (ONOS) [190], Central Office Re-architected as a Datacentre (CORD) [168], O-RAN [167], ONAP [195], Aether [169], and SD-RAN [170].

NFV supports scalable and flexible management and orchestration capabilities [240]. This is achieved by virtualizing network services and functions and separating them from their underlying physical infrastructure. Each feature or functionality is implemented in software format via a virtual network function (VNF) that runs on a virtual machine (VM) instantiated on commercial off-the-shelf (COTS) hardware. Network functions (NFs) such as network address translation (NAT), firewalls, intrusion detection, domain name services, can all be implemented on COTS, reducing the total cost of ownership (TCO) for mobile operators. Network virtualization which began under virtual machines has continuously evolved through containerization and most recently to a cloud-native network architecture.

5.4.2. *Hypervisors, containers, virtual machines*

A hypervisor is a software layer that separates an operating system and application from the underlying physical infrastructure and allows the underlying host machine to support multiple guest VMs. The hypervisor may interconnect several SDN providers under a single abstraction, allowing applications to create E2E slices without having to observe the differences between SDN providers. The network slice constitutes a variety of virtual machines connected by (a) a virtual local area network (VLAN) if the VMs are hosted in the same data centre (DC) or (b) VLAN and Generic Routing Encapsulation if the VMs are in different DCs. The VM interacts with the underlying physical hardware using the hypervisor. Network slicing has been investigated in numerous studies related to network hypervisors, including OpenSlice [179], MobileVisor [180], RadioVisor [181], and HyperFlex [182]. Containers are based on the concept of virtualization at the operating system level and are an alternative to hypervisor-based virtual machines [183]. Containers contain the application, dependencies, and even the version of the operating system. This allows nomadic operations of applications on computer systems, or even the cloud. The orchestration and management of different containers to all function as one system is critical to a cloud native architecture. Docker [184], LinuxVserver [185], OpenVZ [186], and Oracle Solaris Containers are all examples of container-based virtualization solutions. Containers can run VNFs in chains to provide flexible 5G network services or applications and lay the foundation for 5G network slicing. A summary of 5G network slicing key technology enablers is provided in Fig. 13 for the reader's reference.

5.5. *Summary and lessons learned*

5G consumers are destined to enjoy access to huge data rates, low latency. Its capability to connect a multitude of devices and a plethora of use cases is good news for the enterprise market. At the centre of this vision is network slicing and virtualization technologies. Network slicing helps InP to create virtual/logical networks on shared physical infrastructure, thereby reducing TCO. Its combines virtual and physical networking capabilities together with programmable and non-programmable technologies create and effectively manage E2E slices. SDN and NFV are critical enablers for E2E network slicing. They support the softwarization and programmability required to support a fully functioning slice. Both 5G SA and NSA supporting several slicing mechanisms including NSSAI, PLMD-ID, SPID etc. Lastly, Preliminary research on shows that network slicing will continue to play a critical role in the realization and monetization of beyond 5G/6G networks.

6. Network slicing management and orchestration

In the NGMN project [10], Network Slice Instances (NSIs) are created using physical or logical resources (network, compute, storage) that are completely or partially isolated from other resources. E2E network slices can belong to one or more administrative domains distributed throughout the entire network. Close interaction with the NFV management and orchestration (MANO) framework to empower the slice lifecycle management and resource orchestration procedures is required. The next subsection briefly introduces both single-domain and multi-domain management and orchestration.

6.1. *Single domain management and orchestration*

The Open Network Foundation [190] defines orchestration as a continuous process of selecting and using the best resources to address

Table 7
Summary of Open source NFV MANO platforms. The X symbol signifies underlying enablers.

Project	Project Owner	Technology enablers			NFV MANO Framework				Multisite
		SDN	NFV	Cloud	NFVO	VIM	NVFM	OSS/BSS	
ONAP [195]	Linux Foundation	X	X	X	X	X	X	X	X
OSM [196]	ETSI	X	X	X	X	X	X		
Cloudify [197]	GigaSpace		X	X	X		X		
OpenBaton [198]	Fraunhofer		X	X	X	X	X		X
X-MANO [199]	H2020 Vital		X		X				X
Gohan [200]	NTT Data	X	X	X	X		X	X	
Tacker [201]	OpenStack		X	X	X		X		
TeNor [202]	FP7 T-NOVA	X	X	X	X				
CORD/XOS [203]	ON.lab	X	X	X	X		X		X

the needs of the customer services. RAN, transport, and 5GC slices, i.e., subnetwork slices for a single domain, can be combined to form a complete virtual mobile communication network. In the context of ETSI NFV, an E2E MANO module is introduced to translate the business models, use cases, and application requirements of slice tenants into NFs. After all relevant configurations have been performed, the slice is finally assigned the infrastructure resources by the MANO module. It also provides APIs for third parties, such as mobile virtual network operators (MVNOs), to manage and manipulate their own slices. The ETSI NFV MANO Industry Specification Group (ISG) provides some principles and guidelines for a MANO architecture in Ref. [187] and summarizes work in a reference architecture depicted in Fig. 14. This framework includes all managers, repositories, and reference points needed to exchange information, manage multidimensional resources across VNFs.

The NFV MANO is made up of the following managers: (a) virtualization infrastructure manager (VIM) responsible for managing the life cycle of NFV infrastructure (NFVI) resources such as computing, network, and storage in a single administrative domain. Examples of commercial VIM solutions are OpenStack [191], Amazon Web Services (AWS) [188], OpenDayLight [189], and ONOS [190], (b) virtualized network function manager (VNFM) responsible for managing the life-cycle of VNFs and their fault, configuration, accounting, performance and security (C) NFV orchestrator (NFVO) managing E2E resource allocation and utilization, (d) Operation/Business Support System (OSS/BSS) incorporates all functionality that the InP requires to ensure successful day today business operations. Successful integration of the ETSI NFV MANO architecture into the OSS/BSS via an open application programming interface (API) guarantees compatibility with conventional system architectures. Several reference points such as Or-vnfm exist to interconnect the ETSI NFV MANO functional blocks. The MANO system is responsible for instantiating migrating and scaling VNF/NS instances, operating and managing network resources and networks slices based on predefined QoS policies.

6.2. Multi-domain management and orchestration

As 5G network slicing gains momentum on the commercial stage, several challenges need to be addressed. For example (a) network technologies that belong to independently managed and operated domains within the same operator's network. Multi-Operator Core network (MOCN 3GPP TS 23.251) is an example of such a scenario. (b) The need for interaction between domestic or international roaming or cellular network operators due to deficiencies in coverage, borders, and territorial sovereignty. Typical example is a world cup broadcast to a country that requires traversing multiple infrastructures belonging to different service provider domains, (c) services that can only be realized by collaboration between multiple players in the NSI, common in mobile communication networks with multi-vendor equipment suppliers.

In the context of ETSI NFV ISG guidelines in Ref. [192], MANO procedures plan and use physical and virtual network resources to create an NSI across multiple administrative domains. The authors in Ref. [117] provided more clarity on the multi-domain concept from multiple perspectives including (a) multi-technology which is stitching together heterogeneous technology domains such as RAN, CN to create and E2E SDN driven platform to accommodate various 5G services and (b) multi-operator perspective i.e., stitching together and managing the

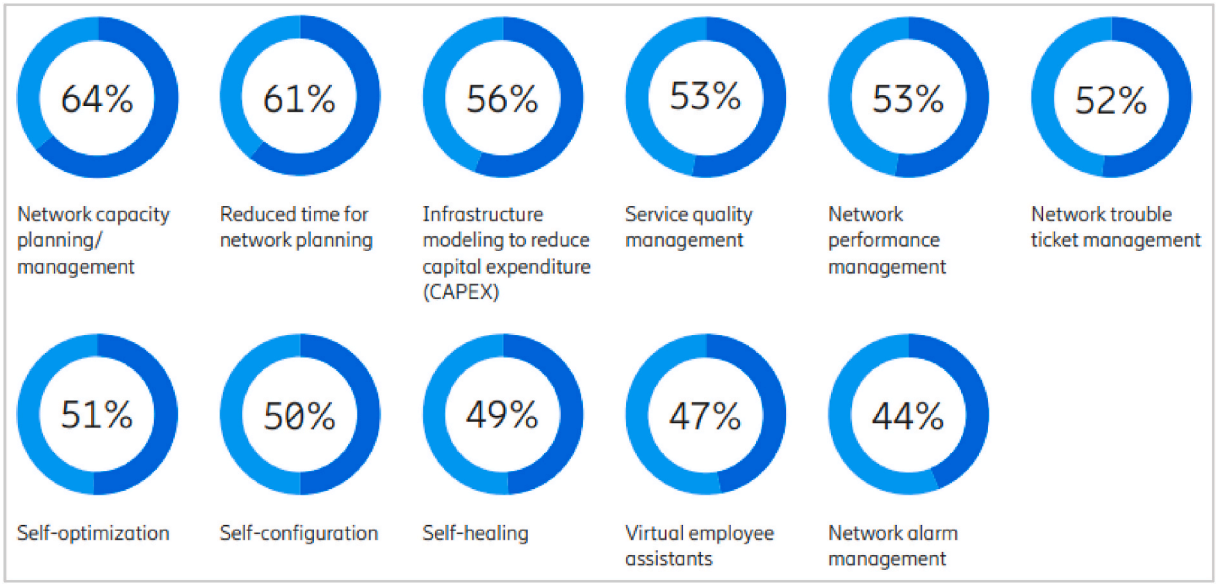


Fig. 15. Areas where service providers will be focusing upon adopting AI in their networks.

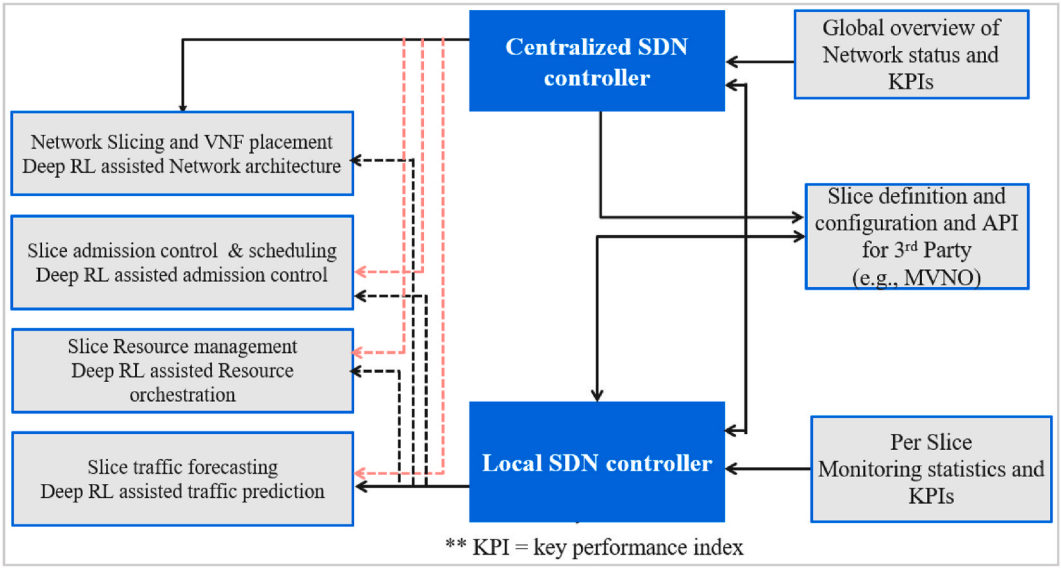


Fig. 16. Functional architecture of Deep RL multi-domain network slicing architecture.

interaction between several InPs or service providers to provide an E2E network slice. The authors in Ref. [150] agree with ETSI NFV ISG in Ref. [192] and argue that to ensure network security, different domain InPs follow their own internal management protocols without disclosing sensitive information to other players in the chain. Mechanisms that enable efficient resource allocation and utilization across multiple administrative domains are generating huge research interest.

The 5GEx framework in Ref. [112] presents an architecture that includes a multi-domain resource orchestrator that is connected to one or more single domain orchestrators. It's the job of the administration domain to manage and steer the multi-domain resource orchestrator. Third-party consumers can access the system via customer-to-business (c2b) interface that is administered by the multi-domain resource orchestrator. Similarly, the 5G!Pagoda framework in Ref. [193], is influenced by the ETSI NFV architecture, with each technology domain having its own resource orchestrator. In addition, management domain resource orchestrators are considered to optimize resource usage and management across multiple administrative domains. The architecture in Ref. [111], also influenced by the ETSI NFV MANO, provides support for multi-domain multi tenancy slicing. Its authors argue for the creation of multi-domain slices as a chain of single domain slices, orchestrated by a single orchestrator to mitigate challenges with direct cross domain orchestration.

The work in Ref. [179] has an inter-slice resource broker that allows sharing resources between slices, and reserving resources based on SLAs between multiple InPs. Whereas several studies only focus on multi-domain MANO architectural challenges, they also address the economic benefits of network slicing across multiple domains. However, their unified management and orchestrator interacts with the inter-slice resource broker and ignores service federation across multiple administrative domains, that may be required to assemble E2E multi domain slice instances. The authors in Ref. [232] address this limitation by presenting a hierarchical multi-domain MANO architecture to address challenges of federated network slicing. Their multi-domain orchestrator is based on recursive abstraction and resource aggregation techniques. The orchestrator aggregates heterogeneous NSI resources first at the domain level and then between federated domains.

Table 8
A summary of the benefits achieved by AI driven solutions over legacy solutions for sliced network management [211].

Function	Perfomance metric	%improvement – benchmark use case
Network slice admission control	Revenue improvement	−0.23%- Optimal ratio 1 −3.77%- Optimal ratio 20 33.3% - Random policies, ratio 15
Cloud resource allocation	Reduction of monetary cost	81.6% - Facebook, core data centre 59.2% - Snapchat, MEC data centre 64.3% - Youtube, C-RAN data centre
Virtual RAN resource allocation	CPU savings, delay, throughput	30% - CPU savings over CPU-blind schedulers 5% - Delay-based QoS over CPU-blind schedulers 25% - Throughput upon computational compacity deficit

6.3. *ETSI NFV MANO open source implementations*

Whereas ETSI [194] presents an NFV MANO reference framework, it doesn't specify its actual implementation. As a result, several open source implementation projects with their underlying orchestration enablers have emerged. Some of them are presented here and they include ONAP [195], OSM [196], Cloudify [197], OpenBaton [198], X-MANO [199], Gohan [200], Tacker [201], TeNor [202], CORD/XOS [203] and many others. Table 7 summarizes open source projects related to ETS NFV MANO and detailed descriptions can be found in Ref. [203].

6.4. *Summary and lessons learned*

The ETSI NFV MANO ISG continues to provide guidelines for managing and orchestrating network service resources across single and multiple domains. Several open source ETSI NFV MANO implementation projects with underlying orchestration enablers continue to emerge to support their work. Network slicing supports multiple applications and use cases with different RAN architectures and QoS requirements that require scalable and flexible network slices. Whereas 5G network slicing and virtualization brings several benefits to InPs, it also introduces several complexities in network operations and management. The good news is that intelligent tools and methods powered by AI/ML provide potential solutions to address these challenges.

7. AI/ML use in 5G network slicing and virtualization

This section describes the important role of AI/ML solutions such as Deep RL in managing and operating network slicing in 5G and beyond

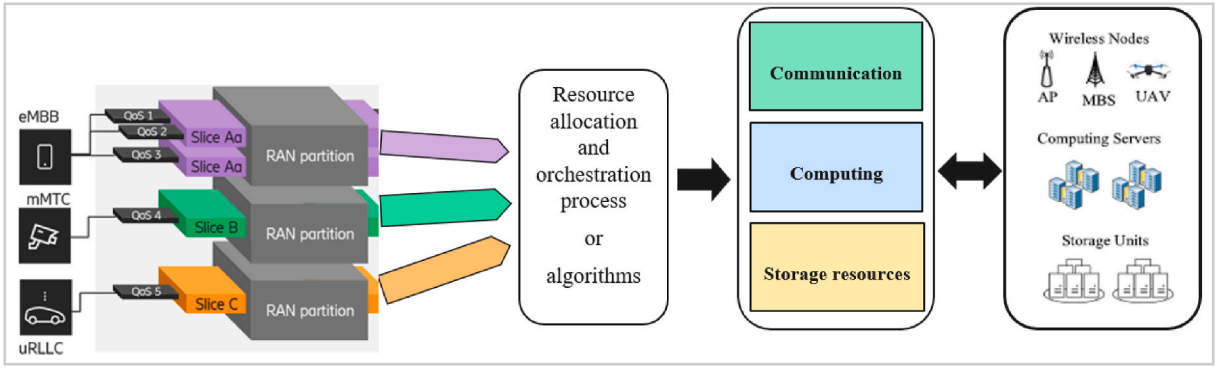


Fig. 17. Diagrammatic representation of resource allocation and orchestration.

networks. AI/ML-based solutions are diverse and will occur at multiple levels, including slice traffic forecast and prediction in all domains, allocating resources to slices in all network domains, controlling the admission of new slices at the RAN, placing slices, reconfiguring slices, managing slice mobility and security etc.

7.1. Overview

As 5G network slicing continues to transform the vertical industries and enterprises, several complexities in network operations, management, fulfilment, and assurance have emerged. Incorporating AI/ML solutions in 5G and beyond network slicing will potentially yield benefits such as improved performance and faster convergence in the network management automation and performance optimization of large-scale systems. According to Ref. [214], the global AI market size is projected to reach USD 309.6 billion by 2026, mainly driven by the need to achieve a robotic economy. The Ericsson report in Ref. [107] outlines key AI/ML opportunities in Fig. 15 for service providers. It notes that 56% of the service provides will adopt AI with a focus on infrastructure modelling to reduce TCO. This reverberates well with the motivation for network slicing. As a result, AI/ML solutions such as Deep RL provide a compelling business case to tackle challenges in 5G network slicing operations. In this section, we review some studies using Deep RL to enable intelligent features and automation 5G network slice topologies, resource allocation and management, admission control, traffic forecasting and prediction.

7.2. Deep RL in the network slicing topology

The introduction of AI tools and methods in slice admission control, resource allocation and utilization, traffic prediction, and VNF placement, functioning in a multi-domain management and orchestration topology, continues to generate excitement in the research community. As discussed earlier, various studies in Refs. [193,232] recommend a tier/hierarchical system topology with an SDN controller in a central cloud and local SDN controllers in each domain, as shown in Fig. 16, to realize slice management and orchestration in multiple administrative domains. When logical nodes are implemented, the SDN controller can be used to support logical link connections between logical nodes that can span multiple administrative domains and several physical nodes in an E2E slice network. The centralized and local SDN controllers could also benefit from Deep RL driven data analysis and decision making in that regard to simplify E2E network management. The network slicing MANO layer creates slices on request. As a result, the successful cooperation between the functional blocks ensures that E2E slice lifecycle management is implemented. Whereas the slice SLA monitoring, resource utilization and traffic prediction data can be used by Deep RL to simplify slice creation, slice admission, slice reconfiguration decision making, statistical model-based techniques become intractable in

Table 9
Summary of RL/Deep RL in slice resource allocation and management.

Authors	Approach/ Solution	Main objectives	Key description
Vassilaras [228]	Model free RL RL with Jacobian-matrix approximation	Resource management	To train an RMM agent as slice manager
Li et al. [229]	Deep RL	Resource management	Tackle radio resource slicing and priority-based core network slicing.
Salhab et al. [220]	ML: LR for classification + Prediction & Deep RL	Resource orchestration, Resource prediction, admission control	Apply ML and Deep RL to tackle resource allocation
Guan et al. [233]	Deep RL	Multi-dimensional resource allocation, dynamic slicing, Admission control	Tackle multi-dimensional resource allocation, dynamic slicing, and slice admission control
Messaoud et al. [234]	Deep federated Q-Learning	Dynamic resource management and allocation	Federated Q-Learning to optimize resource management and allocation
Swapna et al. [235]	RL	Resource orchestration	orchestrate slice elasticity requests to comprehend the maximum revenue
Liu et al. [236]	Master problem: Convex Optimization Slave: Deep RL	Resource allocation and management	Resource allocation problem solved using Convex optimization and Deep RL
Kokku [232]	SDN & NFV slice instantiation	Resource orchestration architecture	federated slices across multi domains
Kibalya et al. [230]	Model free RL	Virtual network embedding (VNEP) problem	Partitioning the slice request to the different candidate substrate networks.
Xi et al. [237]	Deep RL	Resource allocation	Real time resource slicing
Rkham et al. [119]	Deep RL & relational graph convolutional neural networks (CNN)	Resource allocation	A combination of Deep RL & relational graph CNN to improve VNEP heuristics
Van Huynh et al. [68]	SMDP framework, using DNN + Q-learning	Joint resource (radio, computing, storage) allocation	Multidimensional - joint resource allocation
Sun et al. [213]	Deep RL	resource allocation and management	autonomous resource management for multiple slices
Li et al. [259]	Deep RL	resource allocation and management	Comparing Deep RL and intuitive solutions in resource allocation for CN and RAN slices

heterogeneous and dynamic environments such as 5G and beyond networks.

Furthermore, according to the NGMN vision [10], network slices are expected to support a plethora of use cases and applications. The versatility of Deep RL tools and methods, presents it as a compelling solution to optimize the management and orchestration of network slices. In Ref. [211], the authors argue for the introduction of AI/ML in the different phases of the E2E slice life cycle, from admission control, dynamic resource allocation and management etc. The expected performance gains are reported to be in the range of 25 and 80% based on representative case studies. They also show that AI-driven network resource allocation and utilization achieves a 50% reduction in monetary costs. A summary of their study is given in Table 8. Similarly, in Ref. [216], the authors present the concept of intelligent slicing in which AI modules are instantiated in a slicing environment and used as needed. Intelligent slices are used to perform a variety of intelligent tasks, giving the InPs the flexibility to incorporate AI algorithms in the network architecture. The authors of [217] propose AI as an integrated architectural feature that allows the elasticity of resources in 5G networks to be used.

The ITU Y.3172 specifications provide guidelines for the adoption of AI/ML in 5G networks and beyond. ITU-T in Ref. [113] drives the adoption of AI/ML solutions in 5G networks and beyond. In the following sub sections, we'll review existing attempts to use Deep RL in various functionalities in slice based 5G and beyond networks. To provide more clarity, we'll review the RL driven solutions as well

7.3. Slice resource allocation and management

In the context of resource theory [219], resource orchestration is defined as "a process that efficiently uses finite resources such as communication, computing, and caching, transforming them into capabilities to improve performance [220]". In the context of 5G network slicing, resource orchestration aims to distribute multidimensional resources to the incoming slices based on the client's SLA [219]. Network resource management includes the management and orchestration of transport, CN and RAN resources such as physical resource blocks (PRBs), virtual resource blocks (vRBs) and VNFs. The E2E network resources are sliced for multi-domain E2E network slicing but with different optimization objectives. Fig. 17 provides a diagrammatic representation of resource allocation and orchestration in a 5G network slicing scenario. Frequently used resource allocation methods include the reservation method and shared based methods discussed in Ref. [222]. In both scenarios, the aim is to maximize overall network utility by adjusting resource allocation according to resource pool constraints and regulatory rules to achieve near-optimal resource allocation. Resource allocation algorithms aim to dynamically slice resources, efficiently allocating them to ensure that one slice doesn't prey on the other. Therefore, the resource orchestration problem can be formulated as an optimization problem aimed at maximizing network utility and satisfying QoS while minimizing TCO.

7.3.1. Slice resource allocation and management challenges

Network slicing is fully entrenched in the planning and build of 5G and beyond networks. A high level of isolation and QoS provisioning is therefore required to share the infrastructure. Consequently, effective resource allocation and programmable management is a prerequisite [223]. Efficient Network resource allocation is difficult to achieve because of several possible reasons: (a) pressure to meet existing SLA requirements while simultaneously maximizing InPs revenue [224], (b) interference between adjacent logical nodes in different virtual networks built on the same physical infrastructure, (c) the need to administer multidimensional resources in a dynamic network environment. Game theory, auction mechanisms, and genetic algorithms are widely used conventional resource allocation mechanisms. They face some limitations as the networks become more dynamic and evolve into denser, heterogeneous installations, that output larger amounts of data. Traditional statistical model-based tools are also widely used in literature, but given uncertain network conditions, the solutions tend to be suboptimal, resulting in inefficient use of resources and increased costs. In addition, the need for a priori traffic profiles impacts their performance in dynamic environments such as 5G and beyond networks. The versatility of AI/ML tools such as model-free Deep RL provides an alternative approach and solutions in addressing these challenges. Hybrid approaches using heuristics, metaheuristics, shallow ML, RL, and Deep RL have shown promising results [119].

7.3.2. Deep RL in slice resource allocation and management

Existing research has shown promising results for Deep RL based solutions in addressing resource allocation and usage challenges in 5G network slicing. To provide extensive clarity, we'll discuss both RL and Deep RL driven solutions in this subsection. The authors in Ref. [229] use Deep RL to address RAN and CN slicing. From their simulation results, they conclude that Deep RL performed better than other conventional solutions. The authors of [228] propose an RL agent to control radio resource management (RRM). The authors in Ref. [230] propose an RL algorithm to enable 5G network slicing across multiple administrative domains. The main constraints considered in their solution include delay and location.

A solution that combines shallow ML and Deep RL is proposed in Ref. [220] to address resource allocation and orchestration in a knapsack problem. The authors of [232] present a management and orchestration architecture for instantiating and managing interconnected network slices. Although no implementation details are provided regarding service requests, they conclude that Deep RL provides superior results to Q-learning and other techniques. In Ref. [233], the authors use Deep RL to manage the slice resource requests and coordinate internal slice resources for each user. A deep RL driven resource allocation scheme is used to guarantee IoT service QoS in Ref. [234]. The authors in Ref. [235] present an RL-based slice management strategy that regulates slice resiliency to maximize InP utility during the life cycle of E2E network slices.

The authors in Ref. [237] combine conventional alternating direct method of multipliers (ADMM) [254] and Deep RL to address a slice resource allocation problem [236]. The main problem is optimized using convex optimization and Deep RL optimizes the sub problem. Considering the real-time resource requirements of slice users, the authors in Ref. [237] formulate the slice resource allocation problem as a semi markov decision process (SMDP) problem and Deep RL is used to optimize resource allocation with the intent to maximize InP's revenue. To solve the challenge of autonomous resource management for multiple slices, the authors in Ref. [213] successfully use Deep RL to adjust the slice allocated resources based actual usage and QoS feedback. Several studies above consider optimizing the procedures for allocating single resources such as radio resources. However, the study in Ref. [68] formulates a multidimensional resource (network, computing, and storage)

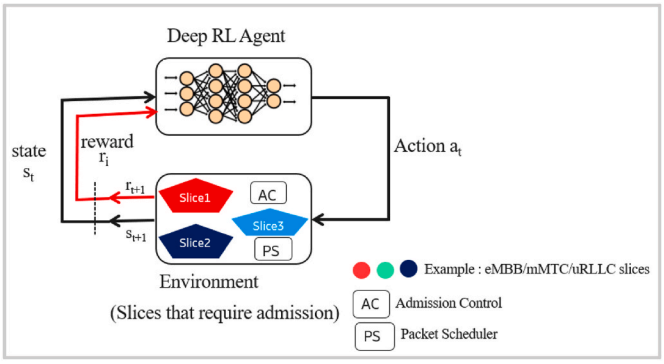


Fig. 18. Adoption of Deep RL in admission control optimization.

allocation problem as an SMDP and Deep RL is used to optimize the resource allocation and utilization accordingly. Their simulations reported 40% improvement in performance. Table 9 provides a summary of the literature review in this sub section. The authors of [259] present a Deep RL driven resource allocation and management solution in RAN and CN slicing. They concluded that Deep RL performs better than other "intuitive solutions" and could also play an important role in next-generation network slicing.

7.4. Slice admission control: overview

Admission control refers to the process of determining the maximum number of concurrent users that a system can allow, bearing in mind the available resources of the wireless system and QoS compliance requirements [238]. When a slice request is received over a 5G network, the slice admission control algorithm determines if the slice can be admitted. Once the slice is approved, an optimization algorithm is used to meet the SLA. This sub-section describes recent efforts to streamline slice admission control using Deep RL. To provide more clarity, we'll review the RL driven solutions as well.

7.4.1. Slice admission control: aims and objectives

Slice admission control (AC) techniques are essential to the efficient management of the 5G network resources [239]. Slice tenants and InP also rely on slice admission control to achieve their predefined goals [240]. This includes maximizing revenue and ensuring end-user QoS and QoE. Whereas available resources in the network resource pool affect slice admittance, the InPs have the choice to admit slices that are most likely to reach a predefined objective. Slice requests are often queued or deferred while the admission algorithm checks for eligibility, considering the available resources. The main objective of slice admission control includes the following: (a) to maximize network utility by optimally distributing network resources among slices. For example, a premium could be charged for low latency and high bandwidth slices to maximize network utility, (b) to control QoS and QoE [239] by accepting only slices whose QoS/QoE can be assured to retain the availability of resources for QoS-intensive network slices. A slice tenant prioritizing the admission of slices serving a world cup venue in a contracted period to guarantee viewers' QoS/QoE at the expense of the other slices as an example, (c) to control congestion [241] by avoiding clogging the system with slices that have a higher probability of rejection such as low priority slices and (d) guarantee fairness [242]: to admit slice requests in a reasonable manner, ensuring that no slice type is frequently admitted at the cost of other slice requests [239].

Strategies such as round-robin queueing and multi-queueing admission control detailed in Ref. [242] work effectively to ensure fairness.

Table 10
Summary of admission control and scheduling approaches in Literature.

Authors	Admission control objective	Mathematical Modelling	Admission strategy	Optimization algorithm
Bega et al. [256]	Revenue maximization	SMDP	Priority based	Deep RL
Yan et al. [258]	efficient resource allocation and utilization	unspecified	optimal	DL and Deep RL
Bakri et al. [260]	utility maximization	unspecified	Optimal	RL, Deep RL, Regret Matching
Raza et al. [262]	High revenue, low degradation	Stochastic policy network	Priority based	Deep RL
Bouzidi et al. [263]	optimizing network latency in an SDN	unspecified	Optimal	Deep RL
Vincenzi et al. [264]	utility maximization	Economic/ Game model	Optimal	RL
Sciancalepore et al. [71]	Efficient resource allocation and utilization	Knapsack problem	Optimal	RL

7.4.2. Slice admission control and optimization: strategies

The commonly adopted slice admission control strategies include the following: (a) first come first served (FCFS), aims to admit slices as they come, if resources are available, as per admissibility design [249], without considering prioritization or QoS indices such as latency and bandwidth [248], (b) priority based admission, commonly referred to as "earliest deadline first" aims to admit high margin priority slices first. Typical examples are slices serving remote eHealth and autonomous driving applications. The challenge for the InP is to strike a balance between prioritizing such high revenue but occasional short term slice contracts vs the low margin but long term contracted slices, (c) greedy based admission is intended to admit slices if capacity constraints are not violated [243], (d) random admission to eliminate unfairness by achieving a fair normal distribution over time [239] and (e) optimal admission uses a policy to accept or deny slices based on acceptable network KPIs, under specific constraints and predefined objectives. Commonly used methods in this context include ML [220], RL [251], Deep RL [251], successive convex approximation (SCA) [253,254], and ADMM [254].

7.4.3. Slice admission control challenges

As often emphasized, network resources (i.e., network, computing, storage) are usually limited. Therefore, to equitably allocate the limited resources, the InPs must make difficult decisions about which slices should be admitted or rejected accordingly. It's always difficult to find the best policy to balance accepting many slices to maximize InP's network utilities while meeting SLAs. Conventional methods such as statistical models, have been used to solve admission control challenges. However, they often require a prior traffic profiles that may not be readily available in various dynamic network environments such as 5G networks and beyond. As a result, model-free RL, Deep RL solutions provide a compelling alternative solution.

7.4.4. Deep RL in slice admission control

In slice admission control, the result of an action is a binary value that indicates whether to allow or deny the slices. Whereas some studies have recommended outright rejection of the slices that can't be admitted, overbooking [257] and other approaches to improve QoS/-QoE and to maximize network utility have been proposed. Some studies have also attempted to incorporate AI/ML technologies into slice admission control to improve efficiency. Fig. 18 shows a summary of the slice admission control process using Deep RL. The learning agent interacts with the environment (usually the admission requirements parameters) in a trial and error manner, receiving a reward in the process that is based on the action taken. Based on the value of the reward obtained for admitting or rejecting the slice, the agent can improve the next course of action. In Ref. [256], the authors design an admissibility region for slices. Using a priority based admission strategy, they use Deep RL to optimize their slice admission control. Their simulations show that the algorithm converges faster and revenue improvements reach 100% in some instances.

The authors in Ref. [260] compared Q-Learning, Deep Q-Learning, and Regret Matching in slice admission control with the objective of maximizing InP utility and understanding the benefits of training the agent offline and online. They conclude that in non-complex problems with small state-action spaces, regret matching performs well. However, in complex problems with high order spaces, Deep Q-Learning is the best option. They also conclude that offline solutions require a training period before use and are costly. However, they usually give the best

results. In Ref. [264], the authors use RL to optimize priority slice admission control to maximize network utility. Services that are likely to generate high revenue with less KPI degradation are admitted based on the learned policy. When compared to two deterministic heuristics, the performance of RL-based admission policies performs better by 55%.

The authors in Ref. [247] considered the threshold policy approach and a combined RL-NN based policy approach. Their simulation results show that the RL policy outperforms the threshold based policies in the scenario with heterogeneous time varying arrival rates and multiple user device types. The authors of [262] propose an RL driven slice admission approach to maximize network utility by learning which slices should be admitted or rejected. The RL solution beats the benchmark heuristics by 55% in performance. Table 10 provides a summary of the literature review in this sub section. In Ref. [71], the authors design a slice broker to support slice admission with the final objective to optimize system resource allocation and utilization. They use RL methods to solve a knapsack problem and optimize the slice admission control. In the same study, they also demonstrate the impact of effective slice traffic forecasting and prediction on resource allocation and utilization. They conclude that resource utilization gains increase with less SLA violations, and effective slice traffic forecast improves resource utilization. Their study demostrates the benefits of incorporating Deep RL in various stages of the slice life cycle management including slice admission control , resouce callocation and traffic forecast and prediction.

7.5. Slice traffic prediction

Predicting wireless data traffic over 5G networks is an important factor in proactive network optimization [252]. Slice traffic can be uncertain, or even fluctuate due to unexpected events like weekend festivities, resulting in unoptimized resource usage, poor QoS/QoE, and, as a result, SLA violations. By accurately predicting the actual footprint of network slices, the InPs can increase the maximum number of slices that fit the same infrastructure [244] and prevent penalties. A detailed account of the objectives of this functionality will be described in the following sub-section. We also review recent efforts to streamline slice traffic forecasting and prediction using Deep RL. To provide more clarity, we'll review the RL driven solutions as well.

Table 11
Summary of the reviewed slice traffic forecasting and prediction studies.

Authors	Objective	Traffic forecasting or prediction strategy	
		Heuristic/ metaheuristic/ Statistical models	RL/DRL Tools
Sciancalepore et al. [71]	Predict slice traffic to improve admission control, resource utilization	Holt-Winters (HW)	RL
Bega et al. [249]	Predict slice traffic to optimize admission control/& revenue	unspecified	RL, Q-learning
Khatibi et al. [209]	Improve resource utilization through slice traffic prediction	unspecified	DNN
Bouzidi et al. [263]	traffic prediction for flow routing optimization	unspecified	Deep RL
Troia et al. [7]	Improve Traffic engineering for utility maximization	unspecified	Deep RL, Policy gradient, TD-λ Deep Q-Learning
Chichali et al. [8]	Traffic prediction, to improve admission and scheduling	Heuristic	RL

7.5.1. Slice traffic prediction objectives

The quest to improve network utility and improve slice resource utilization is at the forefront of the slice traffic forecasting and prediction drive. Forecasting and predicting slice traffic helps the InPs achieve several objectives including the following: (i) satisfy high priority slices through the allocation of more radio resources and throughput [208] (ii) minimize understocking by avoiding to allocate each slice with less than the expected throughput due to less resources to meet throughput needs of all slices [23], (iii) minimize SLA violations by allocating enough resources to each network slice to meet the requirements as per SLA. As described in Ref. [23], SLA categories include Guaranteed Bitrate, Best effort with minimum Guaranteed Bitrate (BEBG), Best Effort (BE), and Guarantee Fairness. (iv) Guarantee fairness [23] i.e resources allocated to each slice should satisfy the predicted demand when network resources are adequate. Any violation and understocking penalties resulting from network congestion will be shared across slices according to their priority to guarantee fairness.

7.5.2. Slice traffic forecasting and prediction challenges

Forecasting and prediction methods are often chosen with some metrics in mind, such as computational cost, average error reduction, and traffic behaviour. The most common traffic forecasting and prediction methods are the conventional linear statistical models ARMA [205] and its derivatives, HoltWinter [206], and non-linear methods based on shallow and deep neural networks [207]. Accurate traffic forecasting and prediction is important for stable, high quality 5G network slicing performance. However, as discussed in Ref. [154], traditional statistical models face some challenges that often impact the performance of network slicing. Here are some examples: (a) Linear methods such as ARIMA are less robust to complex temporal fluctuations such as traffic generation and long term. This is because the model tends to over reproduce the average of previously observed instances [154]. In nonhomogeneous time series scenarios where the inputs and predictions are not in the same set of data points, these methods tend to perform poorly, (b) The general presumption that the slice should always be allocated with the maximum throughput as indicated in the SLA or that throughput should be assigned based on requirements of the admitted slice without any control mechanisms, (c) the accuracy of the results could be impacted by any shifts in the population, (d) supporting seasonal data i.e. time series data with a repeating cycle such as days of the week could be challenging for traditional methods like ARIMA. Accurately predicting slice footprint, results in increased number of slices on the common physical infrastructure [71]. However, today most techniques in traffic forecasting are mainly timeseries methods that ignore the spatial impact of traffic networks in traffic flow modelling [154]. Consideration of spatial and temporal dimensions in traffic forecasting is the key to comprehensive traffic forecasting. While traditional approaches cannot capture these characteristics, AI/ML methods such as Deep RL provide alternative solutions. The following sections describe existing attempts to include Deep RL in slice traffic forecast and prediction. To provide more clarity, we'll review the RL driven solutions as well.

7.5.3. Deep RL in traffic forecasting and prediction

Forecasting, and predicting slice traffic improves slice admission control, resource allocation and usage, slice placement, and other areas of slice lifecycle management. Some studies use traditional linear statistical models to forecast and predict traffic. However, the application of AI/ML in this area is gradually being accepted given the challenges and limitations of the traditional methods discussed earlier. In Ref. [209], the authors use AI approaches to investigate trends in traffic requirements for each network slice and uses empirical information to make predictions. Based on the predictions, an elastic radio resource management model is designed to improve the multiplexing gain without affecting slice QoS. The addition of a DNN to the resource management algorithm to predict resource needs, improves resource

provisioning and use. Their actual evaluation shows that the elastic resource management capabilities of the DNN driven solution optimizes resource usage. A closed-loop solution is proposed in Ref. [71] to address dynamic services such as user network resource requirements. It predicts future slicing requirements to optimize resource usage. Like the [37] approach, they used the HoltWinters approach [206] to predict network slice requirements by tracking past traffic and user mobility. Several control parameters are used to influence the prediction. In addition, only short-term rewards are considered. InP long-term rewards are ignored. The authors in Ref. [249] address this limitation using RL. By using a Semi Markov Decision Process (SMDP) problem formulation approach, a Q-learning-based adaptive algorithm is developed that handles the approval and allocation of network slice requests to maximize InP revenue and guarantee tenant SLAs.

In [7], the Deep RL algorithm is used to address the limitations of traditional traffic engineering solutions. In this study, the policy gradient, TDλ Deep Q-Learning are all implemented. The results show that Deep RL agents with well-designed rewarding capabilities perform best in utility maximization. In Ref. [8], the authors introduce a scheduler to dynamically adapt to traffic fluctuations and various reward functions for optimal planning of IoT traffic. Their results show that the RL scheduler is superior to the heuristic scheduler. The authors of [204] propose a DQN algorithm to overcome the challenges of forecasting traffic demand in mobile traffic offload schemes. By training base station agents using open source datasets, their simulations show that the DQN returns better accuracy in traffic prediction when compared to Q-Learning. Table 11 provides an overview of the reviewed slice traffic forecasts and prediction studies. The authors in Ref. [263] propose a Deep RL approach for predicting traffic and improving the flow routing in an SDN network context. They aimed to collect optimal paths from the Deep RL agent and use deep neural networks to predict future traffic requirements to improve the flow routes. The Deep RL agent performed better than the ARIMA model.

7.6. Summary and lessons learned

This section explored existing efforts to address some key challenges in slicing and virtualizing 5G and beyond networks using DeepRL. It was clear that balancing the allocated slice resources with the needs of actual users was one of InP's biggest challenges. Many studies have also shown that maximizing InP revenue influences many slice admission decisions. Various heuristics, metaheuristics, AI/ML techniques, and statistical model-based solutions have been proposed to automate or optimize this process. However, when the network is congested and resources limited, it's still challenging to find the best policies that can balance the acceptance of multiple slices to maximize InP revenue while meeting SLA requirements. The performance of linear statistical models under uncertain and unstable slice traffic conditions is often suboptimal. Algorithms that predict slice traffic so that the optimal resources are allocated based on actual needs and not SLAs or priority are very desirable. Forecasting and predicting slice traffic helps improve resource allocation and streamlines the slice admission process. Some studies address the problem of forecasting slice traffic and forecasting using traditional linear statistical model-based solutions, especially Holt-Winters. However, if a priori traffic profiles are required, they often impact performance in heterogeneous dynamic environments such as 5G and beyond. ML tools and methods such as Deep RL have been used for decision making, but its use in traffic forecasting and prediction is still limited. Traditional linear statistical models are still dominant.

Hybrid methods that combine Deep RL with heuristics and metaheuristics have also been proposed to accelerate learning and convergence with promising results. Pertinent but not the focus of our study were issues related to inter and intra slice isolation, slice security, slice placement and mobility that have also been widely studied in literature. Overall, pertinent to our study objective, it's clear throughout the reviewed literature that slice resource allocation and utilization, admission control and scheduling, and traffic prediction are critical to effective slice lifecycle management. Addressing all relevant challenges enables effective 5G dynamic network slicing, guarantees SLAs and QoS, maximizes the revenue of InPs, and improves end-user QoE.

8. Open challenges and directions for future research

Deep RL use in 5G mobile communication continues to generate a lot of excitement in the research community. It's fitting that we end our paper by outlining some of the open problems and challenges that could be of interest to the research community. For completeness's sake, our recommendations are spread throughout the various domains in 5G wireless networking.

8.1. The use of multi-agent deep RL in 5G IoT networks

5G is envisioned to support a plethora of Internet of Things (IoT) devices with most IoT sensors being held by the device owner rather than the mobile network operator. As a result, traditional Deep RL customization which works well for a single network entity may not be appropriate in a heterogeneous, dynamic 5G and beyond network environment with a multitude of players. Interactions between different owners (in this case agents) of IoT devices make managing network resources and services extremely difficult and significantly increase the state-action space in question. This condition inevitably slows down the learning algorithm and jeopardizes the performance of the learning policy. In this regard, more effort is needed to accelerate deep multi-agent RL research.

8.2. Distributed DRL application at the edge

Deep RL requires DQN training. This is done by a centralized computer with high computing and processing capabilities. Centralized Deep RL, on the other hand, does not work well on distributed 5G systems such as large IoT devices. As a result, there is a need to design distributed deep RL solutions that divide resource-intensive DQN training tasks into several smaller tasks for individual devices with moderate or limited computing capabilities. This will enable the realization of Deep RL solutions on commercial narrow band-IoT (NB-IoT) and category M1 (CAT-1) devices.

8.3. Deep RL driven dynamic slice admission control

To maximize InP's utility effective slice admission control and scheduling mechanisms will be very important. The versatile and dynamic capabilities of Deep RL make their choice attractive considering that when a new slice is requested, the defined state and action space must also adapt to changes in the slice state-action space for any changes in the network environment. How to speed up convergence is an added complexity to this problem. Therefore, more research is needed in this regard, especially if Deep RL solutions are to be put to production.

8.4. Deep RL driven slice traffic forecasting and prediction

It's difficult to manage slice resources when there are several network slices with varying SLAs and objectives. There is a need to build algorithms to forecast and predict slice traffic so that the optimal resources are allocated based on actual needs and not SLAs or priority. Although Deep RL methods have been proposed for slice traffic forecast and prediction, there is a paucity of research in that space. Conventional statistical models tend to be dominant.

8.5. Latency and accuracy to retrieve rewards

In some of the solutions presented in literature, the RL/Deep RL agent is expected to obtain a quick and accurate for combination of

state-action pairs. However, in a heterogenous and dynamic 5G network environment, this assumption may not be valid because the mobile device may take a long time to send the information and the network may not receive the information properly. For example, in an extended range coverage scenario, the network receiver (Rx) expects the first signal sent by the mobile device to arrive within a specific time window. If the mobile device signal is received beyond the time window, it may not connect to the network at all. The definition of state-action pairs may also prove cumbersome. Therefore, studies on how to design clear and accurate rewards in this aspect requires more efforts.

8.6. Multidimensional resource allocation

This study revealed that the dynamic and non-uniform nature of 5G and beyond networks makes it difficult to manage slice resources. However, many studies presented show that the authors are considering optimizing only 5G wireless communication resource allocation, ignoring computing and storage resources. This approach may not be the best solution for efficient 5G and beyond network slicing resource allocation. More efforts towards joint multi dimensional resource allocation research is required.

9. Conclusion

Deep RL's role in 5G mobile communication continues to generate a lot of enthusiasm and interest in the research community. This paper described the current efforts to use the Deep RL framework in 5G and beyond network slicing and virtualization. We began by diving into the concepts of AI evolution, RL and Deep RL. We explained the basic principles of 5G network slicing and virtualization, as well as key technology enablers. We described the fundamental principles of slice management and orchestration, as well as the challenges. We motivated for the use of AI/ML tools such as Deep RL to address 5G network slice lifecycle management and orchestration challenges. Deep RL and its use to address slice network topology problems, slice admission control and scheduling challenges, slice resource allocation and management, slice traffic forecasting and prediction problems was the focus of this review paper. Successfully showcasing the importance of AI/ML tools in optimizing the 5G slice lifecycle, with Deep RL framework being a potential contributor was the pinnacle of the paper. We also provided some insights on the benefits of Deep RL in the general 5G mobile networking domain. Problem areas that require further studies have also been presented. Network slicing is central to the realization of the 5G vision. We hope that the content of this paper has provided helpful insights into Deep RL and 5G network slicing and will help inspire further research in this field.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] ITU. Framework of the IMT-2020 network. Tech. Rep. Rec. ITU-T Y. May 2018; 3102.
[2] Zhou X, Li R, Chen T, Zhang H. Network slicing as a service: enabling enterprises' own software-defined cellular networks. IEEE Commun Mag 2016;54(7):146–53.
[3] ONF. Applying SDN architecture to 5G slicing. Tech. Rep. TR- Apr. 2016;526.
[4] Ordonez-Lucena J, Ameigeiras P, Lopez D, Ramos-Munoz JJ, Lorca J, Folgueira J. Network slicing for 5G with SDN/NFV: concepts, architectures, and challenges. IEEE Commun Mag 2017;55(5):80–7.
[5] Sivaganesan D. Design and development ai-enabled edge computing for intelligent-iot applications. Journal of trends in Computer Science and Smart technology (TCSST) 2019;1(2):84–94.
[6] Toosi AN, Mahmud R, Chi Q, Buyya R. Management and orchestration of network slices in 5G, fog, edge, and clouds. Fog Edge Comput., Princ. Paradigms 2019;8: 79–96.
[7] Troia S, Sapienza F, Varé L, Maier G. On deep reinforcement learning for traffic engineering in sd-wan. IEEE J Sel Area Commun 2020;39(7):2198–212.
[8] Chinchali S, Hu P, Chu T, Sharma M, Bansal M, Misra R, Pavone M, Katti S. April. Cellular network traffic scheduling with deep reinforcement learning. In: Thirty-second AAAI conference on artificial intelligence; 2018.
[9] Q. Ye, W. Zhuang, S. Zhang, A. Jin, X. Shen, and X. Li, "Dynamic radio resource slicing for a two-tier heterogeneous wireless network," IEEE Trans Veh Technol, vol. 67, no. 10, pp. 9896.
[10] Alliance NGMN. Description of network slicing concept. NGMN 5G P 2016;1(1).
[11] Gawłowicz P, Zubow A. ns3-gym: extending openai gym for networking research. 2018. arXiv preprint arXiv:1810.03943.
[12] Santos J, Wauters T, Volckaert B, De Turck F. Fog computing: enabling the management and orchestration of smart city applications in 5G networks. Entropy 2018;20(1):4.
[13] Wang M, Lin Y, Tian Q, Si G. Transfer learning promotes 6G wireless communications: recent advances and future challenges. IEEE Transactions on Reliability; 2021.
[14] Li X, Fang J, Cheng W, Duan H, Chen Z, Li H. Intelligent power control for spectrum sharing in cognitive radios: a deep reinforcement learning approach. IEEE access 2018;6:25463–73.
[15] Shah S, Dey D, Lovett C, Kapoor A. Airsim: high-fidelity visual and physical simulation for autonomous vehicles. In: Field and service robotics. Cham: Springer; 2018. p. 621–35.
[16] Rohde D, Bonner S, Dunlop T, Vasile F, Karatzoglou A. Recogym: a reinforcement learning environment for the problem of product recommendation in online advertising. 2018. arXiv preprint arXiv:1808.00720.
[17] Silver D, Schrittwieser J, Simonyan K, Antonoglou I, Huang A, Guez A, Hubert T, Baker L, Lai M, Bolton A, Chen Y. Mastering the game of go without human knowledge. Nature 2017;550(7676):354–9.
[18] Silver D, Hubert T, Schrittwieser J, Antonoglou I, Lai M, Guez A, Lanctot M, Sifre L, Kumaran D, Graepel T, Lillicrap T. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. Science 2018;362 (6419):1140–4.
[19] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, Van Den Driessche G, Schrittwieser J, Antonoglou I, Panneershelvam V, Lanctot M, Dieleman S. Mastering the game of Go with deep neural networks and tree search. Nature 2016;529(7587):484–9.
[20] Wydmuch M, Kempka M, Jaśkowski W. Vizdoom competitions: playing doom from pixels. IEEE Trans Games 2018;11(3):248–59.
[21] Lanctot M, Lockhart E, Lespiau JB, Zambaldi V, Upadhyay S, Pérolat J, Srinivasan S, Timbers F, Tuyls K, Omidshafiei S, Hennes D. OpenSpiel: a framework for reinforcement learning in games. 2019. arXiv preprint arXiv: 1908.09453.
[22] He Y, Yu FR, Zhao N, Leung VC, Yin H. Software-defined networks with mobile edge computing and caching for smart cities: a big data deep reinforcement learning approach. IEEE Commun Mag 2017;55(12):31–7.
[23] Gauci J, Conti E, Liang Y, Virochsiri K, He Y, Kaden Z, Narayanan V, Ye X, Chen Z, Fujimoto S. Horizon: facebook's open source applied reinforcement learning platform. 2018. arXiv preprint arXiv:1811.00260.
[24] Delp SL, Anderson FC, Arnold AS, Loan P, Habib A, John CT, Guendelman E, Thelen DG. OpenSim: open-source software to create and analyze dynamic simulations of movement. IEEE Trans Biomed Eng 2007;54(11):1940–50.
[25] Liu J, Krishnamachari B, Zhou S, Niu Z. DeepNap: data-driven base station sleeping operations through deep reinforcement learning. IEEE Internet Things J 2018;5(6):4273–82.
[26] Johnson M, Hofmann K, Hutton T, Bignell D. The Malmo platform for artificial intelligence experimentation. In: Ijcai; 2016, July. p. 4246–7.
[28] Xu X, Zuo L, Huang Z. Reinforcement learning algorithms with function approximation: recent advances and applications, vol. 261. Information Sciences; 2014. p. 1–31.
[29] Kaelbling LP, Littman ML, Moore AW. Reinforcement learning: a survey. J Artif Intell Res 1996;4:237–85.
[30] Ghavamzadeh M, Mannor S, Pineau J, Tamar A. Bayesian reinforcement learning: a survey. 2016. arXiv preprint arXiv:1609.04436.
[31] Mosavi A, Faghan Y, Ghamisi P, Duan P, Ardabili SF, Salwana E, Band SS. Comprehensive review of deep reinforcement learning methods and applications in economics. Mathematics 2020;8(10).
[32] Arora S, Doshi P. A survey of inverse reinforcement learning: challenges, methods and progress. Artificial Intelligence; 2021. p. 103500.
[33] Wirth C, Akrour R, Neumann G, Fürnkranz J. A survey of preference-based reinforcement learning methods. J Mach Learn Res 2017;18(136):1–46.
[34] Qian Y, Wu J, Wang R, Zhu F, Zhang W. Survey on reinforcement learning applications in communication networks. 2019.
[35] Sutton RS, Barto AG. Reinforcement learning: an introduction. MIT press; 2018.
[36] Singh NK, Gupta BM, Dhawan SM, Thakur VK. Reinforcement learning research: a scientometric assessment of global publications output during 2009-18. J Indian Libr Assoc 2021;56(2):27–38.
[37] Zhou J, Zhao W, Chen S. Dynamic network slice scaling assisted by prediction in 5G network. IEEE Access 2020;8:133700–12.
[38] Garcıa J, Fernández F. A comprehensive survey on safe reinforcement learning. J Mach Learn Res 2015;16(1):1437–80.
[39] Hua Y, Li R, Zhao Z, Chen X, Zhang H. GAN-powered deep distributional reinforcement learning for resource management in network slicing. IEEE J Sel Area Commun 2019;38(2):334–49.
[40] Taylor ME, Stone P. Transfer learning for reinforcement learning domains: a survey. J Mach Learn Res 2009;10(7).

[41] Busoniu L, Babuska R, De Schutter B. A comprehensive survey of multiagent reinforcement learning. IEEE Trans Syst Man Cybern C Appl Rev 2008;38(2):156–72.
[42] Kiumarsi B, Vamvoudakis KG, Modares H, Lewis FL. Optimal and autonomous control using reinforcement learning: a survey. IEEE Transact Neural Networks Learn Syst 2017;29(6):2042–62.
[43] Moerland TM, Broekens J, Jonker CM. Model-based reinforcement learning: a survey. 2020. arXiv preprint arXiv:2006.16712.
[44] Padakandla S. A survey of reinforcement learning algorithms for dynamically varying environments. 2020. arXiv preprint arXiv:2005.10619.
[45] Nguyen ND, Nguyen T, Nahavandi S. System design perspective for human-level agents using deep reinforcement learning: a survey. IEEE Access 2017;5:27091–102.
[46] Zhao X, Xia L, Tang J, Yin D. Deep reinforcement learning for search, recommendation, and online advertising: a survey" by Xiangyu Zhao, long Xia, Jiliang Tang, and Dawei Yin with Martin Wesely as coordinator. ACM SIGWEB Newsletter, (Spring); 2019. p. 1–15.
[47] Li Y. Deep reinforcement learning. 2018. arXiv preprint arXiv:1810.06339.
[48] Xiong Z, Zhang Y, Niyato D, Deng R, Wang P, Wang LC. Deep reinforcement learning for mobile 5G and beyond: fundamentals, applications, and challenges. IEEE Veh Technol Mag 2019;14(2):44–52.
[49] Hernandez-Leal P, Kartal B, Taylor ME. A survey and critique of multiagent deep reinforcement learning. Aut Agents Multi-Agent Syst 2019;33(6):750–97.
[50] Zhao X, Xia L, Tang J, Yin D. Deep reinforcement learning for search, recommendation, and online advertising: a survey" by Xiangyu Zhao, long Xia, Jiliang Tang, and Dawei Yin with Martin Wesely as coordinator. ACM SIGWEB Newsletter, (Spring); 2019. p. 1–15.
[51] Yu L, Qin S, Zhang M, Shen C, Jiang T, Guan X. Deep reinforcement learning for smart building energy management: a survey. 2020. arXiv preprint arXiv:2008.05074.
[52] Kiran BR, Sobh I, Talpaert V, Mannion P, Al Sallab AA, Yogamani S, Pérez P. Deep reinforcement learning for autonomous driving: a survey. IEEE Transactions on Intelligent Transportation Systems; 2021.
[53] Talpaert V, Sobh I, Kiran BR, Mannion P, Yogamani S, El-Sallab A, Perez P. Exploring applications of deep reinforcement learning for real-world autonomous driving systems. 2019. arXiv preprint arXiv:1901.01536.
[54] Jovović I, Forenbacher I, Periša M. November. Massive machine-type communications: an overview and perspectives towards 5g. In: Proc. 3rd Int. Virtual Res. Conf. Tech. Disciplines, vol. 3; 2015.
[55] Xiong Z, Zhang Y, Niyato D, Deng R, Wang P, Wang LC. Deep reinforcement learning for mobile 5G and beyond: fundamentals, applications, and challenges. IEEE Veh Technol Mag 2019;14(2):44–52.
[56] Luong NC, Hoang DT, Gong S, Niyato D, Wang P, Liang Y-C, Kim DI. Applications of deep reinforcement learning in communications and networking: a survey. IEEE Commun Surv Tutorials 2019;21(4):3133–74.
[57] Qian Y, Wu J, Wang R, Zhu F, Zhang W. Survey on reinforcement learning applications in communication networks. J Commun Inf Network 2019;4(2):30–9.
[58] Lei L, Tan Y, Zheng K, Liu S, Zhang K, Shen X. Deep reinforcement learning for autonomous internet of things: model, applications and challenges. IEEE Communications Surveys & Tutorials; 2020.
[59] Foukas X, Patounas G, Elmokashfi A, Marina MK. Network slicing in 5G: survey and challenges. IEEE Commun Mag 2017;55(5):94–100.
[60] Jovović I, Forenbacher I, Periša M. November. Massive machine-type communications: an overview and perspectives towards 5g. In: Proc. 3rd Int. Virtual Res. Conf. Tech. Disciplines, vol. 3; 2015.
[61] Zhao N, Liang YC, Niyato D, Pei Y, Jiang Y. December. Deep reinforcement learning for user association and resource allocation in heterogeneous networks. In: 2018 IEEE global communications conference (GLOBECOM). IEEE; 2018. p. 1–6.
[62] Liang C, Yu FR. Wireless network virtualization: a survey, some research issues, and challenges. IEEE Commun Surv Tutorials 2014;17(1):358–80.
[63] Barakabitze AA, Ahmad A, Mijumbi R, Hines A. 5G network slicing using SDN and NFV: a survey of taxonomy, architectures and future challenges. Computer Networks; 2020.
[64] Khan LU, Yaqoob I, Tran NH, Han Z, Hong CS. Network slicing: recent advances, taxonomy, requirements, and open research challenges. IEEE Access 2020;8.
[65] Osseiran A, Boccardi F, Braun V, Kusume K, Marsch P, Maternia M, Queseth O, Schellmann M, Schotten H, Taoka H, Tullberg H. Scenarios for 5G mobile and wireless communications: the vision of the METIS project. IEEE Commun Mag 2014;52(5):26–35.
[66] Zhang Q, Liang YC, Poor HV. Intelligent user association for symbiotic radio networks using deep reinforcement learning. IEEE Trans Wireless Commun 2020;19(7):4535–48.
[67] Suárez L, Espes D, Le Parc P, Cuppens F, Bertin P, Phan CT. November. Enhancing network slice security via Artificial Intelligence: challenges and solutions. In: Conférence C&ESAR 2018; 2018.
[68] Van Huynh N, Hoang DT, Nguyen DN, Dutkiewicz E. Optimal and fast real-time resource slicing with deep dueling neural networks. IEEE J Sel Area Commun 2019;37(6):1455–70.
[69] 3GPP Release 15. https://www.3gpp.org/release-15.
[70] Feriani A, Hossain E. Single and multi-agent deep reinforcement learning for AI-enabled wireless networks: a tutorial. IEEE Communications Surveys & Tutorials; 2021.
[71] Sciancalepore V, Samdanis K, Costa-Perez X, Bega D, Gramaglia M, Banchs A. May. Mobile traffic forecasting for maximizing 5G network slicing resource utilization. In: IEEE INFOCOM 2017-IEEE conference on computer communications. IEEE; 2017. p. 1–9.
[72] Agostinelli F, Hocquet G, Singh S, Baldi P. From reinforcement learning to deep reinforcement learning: an overview. Braverman readings in machine learning. key ideas from inception to current state. 2018. p. 298–328.
[73] Russell S, Norvig P. Artificial intelligence: a modern approach. 2002.
[74] Jordan MI, Mitchell TM. Machine learning: trends, perspectives, and prospects. Science 2015;349(6245):255–60.
[75] Samuel AL. Some studies in machine learning using the game of checkers. IBM J Res Dev 1959;3(3):210–29.
[76] Widrow B, Hoff ME. Adaptive switching circuits. Stanford Univ Ca Stanford Electronics Labs; 1960.
[77] Hebb DO. The organization of behavior; a neuropsycholocigal theory, vol. 62. A Wiley Book in Clinical Psychology; 1949. p. 78.
[78] Hebb DO. The organization of behavior: a neuropsychological theory. Psychology Press; 2005.
[79] Michie D, Chambers RA. BOXES: an experiment in adaptive control. Mach Intell 1968;2(2):137–52.
[80] Barto AG, Sutton RS, Brouwer PS. Associative search network: a reinforcement learning associative memory. Biol Cybern 1981;40(3):201–11.
[81] Sutton RS. Learning to predict by the methods of temporal differences. Mach Learn 1988;3(1):9–44.
[82] Watkins CJCH. Learning from delayed rewards. 1989.
[83] Kaelbling LP. Hierarchical learning in stochastic domains: preliminary results. In: Proceedings of the tenth international conference on machine learning, vol. 951; 1993. p. 167–73.
[84] Sutton RS. Introduction: the challenge of reinforcement learning. In: Reinforcement learning. Boston, MA: Springer; 1992. p. 1–3.
[85] Yu S, Chen X, Zhou Z, Gong X, Wu D. When deep reinforcement learning meets federated learning: intelligent multitimescale resource management for multiaccess edge computing in 5G ultradense network. IEEE Internet Things J 2020;8(4):2238–51.
[86] Watkins CJ, Dayan P. Q-learning. Mach Learn 1992;8(3–4):279–92.
[88] Koza John R, Koza John R. Genetic programming: on the programming of computers by means of natural selection, vol. 1. MIT press; 1992.
[89] Bellman R. Dynamic programming. Science 1966;153(3731):34–7.
[90] Tesauro G. Temporal difference learning and TD-Gammon. Commun ACM 1995;38(3):58–68.
[91] Konda VR, Tsitsiklis JN. Actor-critic algorithms. In: Advances in neural information processing systems; 2000. p. 1008–14.
[92] Hamari J, Koivisto J, Sarsa H. January. Does gamification work?–a literature review of empirical studies on gamification. In: 2014 47th Hawaii international conference on system sciences. Ieee; 2014. p. 3025–34.
[93] Dayan Peter, Sejnowski Terrence J. TD (λ) converges with probability 1. Mach Learn 1994;14(3):295–301.
[94] Silver D, Lever G, Heess N, Degris T, Wierstra D, Riedmiller M. January. Deterministic policy gradient algorithms. In: International conference on machine learning. PMLR; 2014. p. 387–95.
[95] Ho HN, Lee E. January. Model-based reinforcement learning approach for planning in self-adaptive software system. In: Proceedings of the 9th international conference on ubiquitous information management and communication; 2015. p. 1–8.
[96] Browne CB, Powley E, Whitehouse D, Lucas SM, Cowling PI, Rohlfshagen P, Tavener S, Perez D, Samothrakis S, Colton S. A survey of Monte Carlo tree search methods. IEEE Transactions on Computational Intelligence and AI in games 2012;4(1):1–43.
[97] Mismar FB, Evans BL, Alkhateeb A. Deep reinforcement learning for 5G networks: joint beamforming, power control, and interference coordination. IEEE Trans Commun 2019;68(3):1581–92.
[98] Vermorel J, Mohri M. October. Multi-armed bandit algorithms and empirical evaluation. In: European conference on machine learning. Berlin, Heidelberg: Springer; 2005. p. 437–48.
[99] Ferns N, Panangaden P, Precup D. Metrics for Markov decision processes with infinite state spaces. 2012. arXiv preprint arXiv:1207.1386.
[100] Tokic M, Palm G. October. Value-difference based exploration: adaptive control between epsilon-greedy and softmax. In: Annual conference on artificial intelligence. Berlin, Heidelberg: Springer; 2011. p. 335–46.
[101] Bertsekas DP. Dynamic programming and optimal control. third ed., vol. 2. Athena Scientific; 2007.
[102] Sutton RS, McAllester DA, Singh SP, Mansour Y. November. Policy gradient methods for reinforcement learning with function approximation. In: NIPs, vol. 99; 1999. p. 1057–63.
[103] Schmidt R, Chang CY, Nikaein N. December. Slice scheduling with QoS-guarantee towards 5G. In: 2019 IEEE global communications conference (GLOBECOM). IEEE; 2019. p. 1–7.
[104] Castro PS, Moitra S, Gelada C, Kumar S, Bellemare MG. Dopamine: a research framework for deep reinforcement learning. 2018. arXiv preprint arXiv:1812.06110.
[105] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, Graves A, Riedmiller M, Fidjeland AK, Ostrovski G, Petersen S. Human-level control through deep reinforcement learning. Nature 2015 Feb;518(7540):529–33.
[106] Schaul T, Quan J, Antonoglou I, Silver D. Prioritized experience replay. 2015. arXiv preprint arXiv:1511.05952.
[107] Employing AI techniques to enhance returns on 5G network investments. https://www.ericsson.com/en/blog/2019/5/ai-in-5g-networks-report-key-highlights.

[108] Chen CT, Chen AP, Huang SH. July. Cloning strategies from trading records using agent-based reinforcement learning algorithm. In: 2018 IEEE international conference on agents (ICA). IEEE; 2018. p. 34–7.
[109] Schaul T, Quan J, Antonoglou I, Silver D. Prioritized experience replay. 2015. arXiv preprint arXiv:1511.05952.
[111] Kukliński S, Tomaszewski L, Osiński T, Ksentini A, Frangoudis PA, Cau E, Corici M. June. A reference architecture for network slicing. In: 2018 4th IEEE conference on network softwarization and workshops (NetSoft). IEEE; 2018. p. 217–21.
[112] Guerzoni R, Perez-Caparros D, Monti P, Giuliani G, Melian J, Biczók G. Multi-domain orchestration and management of software defined infrastructures: a bottom-up approach. 2016.
[113] ITU-FG ML 5G focus group. https://www.itu.int/en/ITU-T/focusgroups/ml5g/Pages/default.aspx.
[114] Nagib AM, Abou-Zeid H, Hassanein HS. October. Transfer learning-based accelerated deep reinforcement learning for 5G RAN slicing. In: 2021 IEEE 46th conference on local computer networks (LCN). IEEE; 2021. p. 249–56.
[115] Liang E, Liaw R, Nishihara R, Moritz P, Fox R, Goldberg K, Gonzalez J, Jordan M, Stoica I. July. RLlib: abstractions for distributed reinforcement learning. In: International conference on machine learning. PMLR; 2018. p. 3053–62.
[116] O'Donoghue B, Osband I, Munos R, Mnih V. July. The uncertainty bellman equation and exploration. In: International conference on machine learning; 2018. p. 3836–45.
[117] ITU-T FG IMT-2020. Report on standards gap analysis. https://www.itu.int/en/ITU-T/focusgroups/imt-2020/Documents/T13-SG13-151130-TD-PLEN-0208!!MSW-E.docx; Dec 2015.
[118] Osband I, Blundell C, Pritzel A, Van Roy B. Deep exploration via bootstrapped DQN. 2016. arXiv preprint arXiv:1602.04621.
[119] Rkhami A, Hadjadj-Aoul Y, Outtagarts A. January. Learn to improve: a novel deep reinforcement learning approach for beyond 5G network slicing. In: 2021 IEEE 18th annual consumer communications & networking conference (CCNC). IEEE; 2021. p. 1–6.
[120] Almahdi S, Yang SY. An adaptive portfolio trading system: a risk-return portfolio optimization using recurrent reinforcement learning with expected maximum drawdown. Expert Syst Appl 2017;87:267–79.
[121] Xu H, Liu X, Yu W, Griffith D, Golmie N. Reinforcement learning-based control and networking co-design for industrial internet of things. IEEE J Sel Area Commun 2020;38(5):885–98.
[122] Zhang D, Maei H, Wang X, Wang YF. Deep reinforcement learning for visual object tracking in videos. 2017. arXiv preprint arXiv:1701.08936.
[123] Luketina J, Nardelli N, Farquhar G, Foerster J, Andreas J, Grefenstette E, Whiteson S, Rocktäschel T. A survey of reinforcement learning informed by natural language. 2019. arXiv preprint arXiv:1906.03926.
[124] Wang WY, Li J, He X. July. Deep reinforcement learning for NLP. In: Proceedings of the 56th annual meeting of the association for computational linguistics: tutorial abstracts; 2018. p. 19–21.
[125] Practical applications RL in industry. https://www.oreilly.com/radar/practical-applications-of-reinforcement-learning-in-industry/.
[126] Bellemare MG, Naddaf Y, Veness J, Bowling M. The arcade learning environment: an evaluation platform for general agents. J Artif Intell Res 2013;47:253–79.
[127] Brockman G, Cheung V, Pettersson L, Schneider J, Schulman J, Tang J, Zaremba W. Openai gym. 2016. arXiv preprint arXiv:1606.01540.
[128] OpenAI gym. https://gym.openai.com/.
[129] MuJoCo. http://www.mujoco.org.
[130] Beattie C, Leibo JZ, Teplyashin D, Ward T, Wainwright M, Küttler H, Lefrancq A, Green S, Valdés V, Sadik A, Schrittwieser J. Deepmind lab. 2016. arXiv preprint arXiv:1612.03801.
[131] Tassa Y, Doron Y, Muldal A, Erez T, Li Y, Casas DDL, Budden D, Abdolmaleki A, Merel J, Lefrancq A, Lillicrap T. Deepmind control suite. 2018. arXiv preprint arXiv:1801.00690.
[132] Tian Y, Gong Q, Shang W, Wu Y, Zitnick CL. Elf: an extensive, lightweight and flexible research platform for real-time strategy games. 2017. arXiv preprint arXiv:1707.01067.
[133] Tian Y, Ma J, Gong Q, Sengupta S, Chen Z, Pinkerton J, Zitnick L. May. Elf opengo: an analysis and open reimplementation of alphazero. In: International conference on machine learning. PMLR; 2019. p. 6244–53.
[134] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521(7553):436–44.
[135] Bishop CM. Pattern recognition and machine learning. springer; 2006.
[136] Littman ML. Reinforcement learning improves behaviour from evaluative feedback. Nature 2015;521(7553):445–51.
[137] Goodfellow I, Bengio Y, Courville A, Bengio Y. Deep learning, vol. 1. Cambridge: MIT press; 2016. No. 2.
[138] Levine S, Pastor P, Krizhevsky A, Ibarz J, Quillen D. Learning hand-eye coordination for robotic grasping with deep learning and large-scale data collection. Int J Robot Res 2018;37(4–5):421–36.
[139] Li Y. Deep reinforcement learning. 2018. arXiv preprint arXiv:1810.06339.
[140] https://deepmind.com/learning-resources/-introduction-reinforcement-learning-david-silver.
[141] Singh NK, Gupta BM, Dhawan SM, Thakur VK. Reinforcement learning research: a scientometric assessment of global publications output during 2009-18. J Indian Libr Assoc 2021;56(2):27–38.
[142] Hastie T, Tibshirani R, Friedman J. Unsupervised learning. In: The elements of statistical learning. New York, NY: Springer; 2009. p. 485–585.
[143] Murphy KP. Machine learning: a probabilistic perspective. MIT press; 2012.
[144] James G, Witten D, Hastie T, Tibshirani R. An introduction to statistical learning, vol. 112. New York: springer; 2013. p. 18.
[145] Domingos P. A few useful things to know about machine learning. Commun ACM 2012;55(10):78–87.
[146] Baylor D, Breck E, Cheng HT, Fiedel N, Foo CY, Haque Z, Haykal S, Ispir M, Jain V, Koc L, Koo CY. August. Tfx: a tensorflow-based production-scale machine learning platform. In: Proceedings of the 23rd ACM SIGKDD international conference on knowledge discovery and data mining; 2017. p. 1387–95.
[147] Ng A, Jordan M, Weiss Y. On spectral clustering: analysis and an algorithm. Adv Neural Inf Process Syst 2001;14:849–56.
[148] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. 2017. arXiv preprint arXiv:1707.06347.
[149] Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D. Continuous control with deep reinforcement learning. 2015. arXiv preprint arXiv:1509.02971.
[150] Mano T, Inoue T, Ikarashi D, Hamada K, Mizutani K, Akashi O. Efficient virtual network optimization across multiple domains without revealing private information. IEEE Trans Network Service Manag 2016;13(3):477–88.
[154] Feng J, Chen X, Gao R, Zeng M, Li Y. Deeptp: an end-to-end neural network for mobile cellular traffic prediction. IEEE Network 2018;32(6):108–15.
[157] Mnih V, Badia AP, Mirza M, Graves A, Lillicrap T, Harley T, Silver D, Kavukcuoglu K. June. Asynchronous methods for deep reinforcement learning. In: International conference on machine learning. PMLR; 2016. p. 1928–37.
[158] Chen S, Liang YC, Sun S, Kang S, Cheng W, Peng M. Vision, requirements, and technology trend of 6G: how to tackle the challenges of system coverage, capacity, user data-rate and movement speed. IEEE Wireless Commun 2020;27(2):218–28.
[159] Zhang Z, Xiao Y, Ma Z, Xiao M, Ding Z, Lei X, Karagiannidis GK, Fan P. 6G wireless networks: vision, requirements, architecture, and key technologies. IEEE Veh Technol Mag 2019;14(3):28–41.
[160] Laborie P, Rogerie J, Shaw P, Vilím P. IBM ILOG CP optimizer for scheduling. Constraints 2018;23(2):210–50.
[161] ITU-T. Terms and definitions for imt-2020 network. Technical report. International Telecommunication Union; 2017.
[162] Ordonez-Lucena J, Ameigeiras P, Lopez D, Ramos-Munoz JJ, Lorca J, Folgueira J. Network slicing for 5G with SDN/NFV: concepts, architectures, and challenges. IEEE Commun Mag 2017;55(5):80–7.
[163] Berman M, et al. GENI: a federated testbed for innovative network experiments. Comput Network Mar. 2014;61:5–23.
[164] Lillicrap TP, Hunt JJ, Pritzel A, Heess N, Erez T, Tassa Y, Silver D, Wierstra D. Continuous control with deep reinforcement learning. arXiv preprint, 1509.02971; 2015.
[165] Taleb T, Mada B, Corici MI, Nakao A, Flinck H. PERMIT: network slicing for personalized 5G mobile telecommunications. IEEE Commun Mag 2017;55(5):88–93.
[166] Gude N, et al. NOX: towards an operating system for networks," ACM SIGCOMM Comput. Commun Rev Jul. 2008;38(3):105–10.
[167] O-RAN. https://www.o-ran.org/.
[168] CORD. Re-inventing central offices for efficiency and agility [Online]. Available: https://opencord.org/. [Accessed May 2021].
[169] https://www.openstack.id/wp-content/uploads/2020/11/Masagung-Nugroho-%E2%80%93-Open-Dissagregated-Router-ID-Rev2.0.pptx.pdf.
[170] SD-RAN. https://opennetworking.org/sd-ran/.
[171] Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. arXiv preprint, 1707.06347; 2017.
[172] Schulman J, Levine S, Abbeel P, Jordan M, Moritz P. June. Trust region policy optimization. In: International conference on machine learning. PMLR; 2015. p. 1889–97.
[173] Denoyer L, de la Fuente A, Duong S, Gaya JB, Kamienny PA, Thompson DH. SaLinA: sequential learning of agents. arXiv preprint, 2110.07910; 2021.
[174] Murugesan K, Atzeni M, Kapanipathi P, Shukla P, Kumaravel S, Tesauro G, Talamadupula K, Sachan M, Campbell M. May. Text-based RL agents with Commonsense knowledge: new challenges, environments and baselines. In: Thirty fifth AAAI conference on artificial intelligence; 2021.
[175] Caron M, Misra I, Mairal J, Goyal P, Bojanowski P, Joulin A. Unsupervised learning of visual features by contrasting cluster assignments. arXiv preprint, 2006.09882; 2020.
[176] Zbontar J, Jing L, Misra I, LeCun Y, Deny S. Barlow twins: self-supervised learning via redundancy reduction. arXiv preprint, 2103.03230; 2021 Mar 4.
[177] Grinsztajn N, Ferret J, Pietquin O, Preux P, Geist M. There is No turning back: a self-supervised approach for reversibility-aware reinforcement learning. arXiv preprint, 2106.04480; 2021.
[178] Barakabitze AA, Ahmad A, Mijumbi R, Hines A. 5G network slicing using SDN and NFV: a survey of taxonomy, architectures and future challenges. Comput Network 2020;167:106984.
[179] Sciancalepore V, Mannweiler C, Yousaf FZ, Serrano P, Gramaglia M, Bradford J, Pavón IL. A future-proof architecture for management and orchestration of multi-domain nextgen networks. IEEE Access 2019;7:79216–32.
[180] Hawilo H, Shami A, Mirahmadi M, Asal R. NFV: state of the art, challenges, and implementation in next generation mobile networks (vEPC). IEEE Network 2014;28(6):18–26.
[181] Pham TM, Chu HN. Multi-provider and multi-domain resource orchestration in network functions virtualization. IEEE Access 2019;7:86920–31.
[182] Guerzoni R, Vaishnavi I, Perez Caparros D, Galis A, Tusa F, Monti P, Sganbelluri A, Biczók G, Sonkoly B, Toka L, Ramos A. Analysis of end-to-end multi-domain management and orchestration frameworks for software defined infrastructures: an architectural survey. Trans Emerg Telecommun Technol 2017;28(4):e3103.

[183] Sattar D, Matrawy A. June. Towards secure slicing: using slice isolation to mitigate DDoS attacks on 5G core network slices. In: 2019 IEEE conference on communications and network security (CNS). IEEE; 2019. p. 82–90.
[184] Khan R, Kumar P, Jayakody DNK, Liyanage M. A survey on security and privacy of 5G technologies: potential solutions, recent advancements, and future. 2019.
[185] An N, Kim Y, Park J, Kwon DH, Lim H. Slice management for quality of service differentiation in Wireless Network Slicing. Sensors 2019;19(12):2745.
[186] Lee MK, Hong CS. July. Efficient slice allocation for novel 5g services. In: 2018 tenth international conference on ubiquitous and future networks (ICUFN); 2018. p. 625–9.
[187] Virtualisation NF. ETSI industry specification group (ISG),"ETSI GS NFV-MAN 001 V1. 1.1. Network Functions Virtualisation (NFV); Management and Orchestration; 2014.
[188] Amazon Web Services (AWS). Cloud computing services [Online]. Available: https://aws.amazon.com/. [Accessed May 2019].
[189] Opendaylight [Online]. Available: https://www.opendaylight.org/. [Accessed May 2021].
[190] Open Networking Foundation. OpenFlow specification. https://www.opennetworking.org/sdn-resources/onf-specifications/openflow/.
[191] OpenStack [Online]. Available: https://www.openstack.org. [Accessed May 2021].
[192] Network Function Virtualisation (NFV). Management and orchestration; report on management and connectivity for multi-site services. 2018.
[193] Afolabi I, Ksentini A, Bagaa M, Taleb T, Corici M, Nakao A. Towards 5G network slicing over multiple-domains. IEICE Trans Commun 2017;100(11):1992–2006.
[194] Han B, Tayade S, Schotten HD. July. Modeling profit of sliced 5G networks for advanced network resource management and slice implementation. In: 2017 IEEE symposium on computers and communications (ISCC). IEEE; 2017. p. 576–81.
[195] Foundation L. ONAP–Open network automation platform [Online]. Available: https://www.onap.org/. [Accessed May 2021].
[196] ETSI. Open source MANO [Online]. Available: https://osm.etsi.org/. [Accessed May 2021].
[197] GigaSpaces. 2015cloudify [Online]. Available: http://cloudify.co/. [Accessed May 2021].
[198] Fraunhofer. Open Baton: an open source reference implementation of the ETSI Network Function Virtualization MANO specification [Online]. Available: http://openbaton.github.io/. [Accessed May 2021].
[199] Francescon A, Baggio G, Fedrizzi R, Ferrusy R, Yahiaz IGB, Riggio R. July. X–MANO: cross–domain management and orchestration of network services. In: 2017 IEEE conference on network softwarization (NetSoft). IEEE; 2017. p. 1–5.
[200] NTT. Gohan-REST-based api server to evolve your cloud service very rapidly [Online]. Available: http://gohan.cloudwan.io/. [Accessed May 2021].
[201] Foundation O. Tacker-OpenStack [Online]. Available: https://wiki.openstack.org/wiki/Tacker. [Accessed May 2021].
[202] Riera JF, Batallé J, Bonnet J, Días M, McGrath M, Petralia G, Liberati F, Giuseppi A, Pietrabissa A, Ceselli A, Petrini A. June. TeNOR: steps towards an orchestration platform for multi-PoP NFV deployment. In: 2016 IEEE NetSoft conference and workshops (NetSoft). IEEE; 2016. p. 243–50.
[203] Mamushiane L, Lysko AA, Mukute T, Mwangama J, Du Toit Z. August. Overview of 9 open-source resource orchestrating ETSI MANO compliant implementations: a brief survey. In: 2019 IEEE 2nd wireless Africa conference (WAC). IEEE; 2019. p. 1–7.
[204] Huang CW, Chen PC. Mobile traffic offloading with forecasting using deep reinforcement learning. arXiv preprint, 1911.07452; 2019.
[205] Yu Y, Wang J, Song M, Song J. October. Network traffic prediction and result analysis based on seasonal ARIMA and correlation coefficient. In: 2010 international conference on intelligent system design and engineering application, vol. 1. IEEE; 2010. p. 980–3.
[206] Koehler AB, Snyder RD, Ord JK. Forecasting models and prediction intervals for the multiplicative Holt-Winters method. Int J Forecast 2001;17(2):269–86.
[207] Sherry J, Lan C, Popa RA, Ratnasamy S. August. Blindbox: deep packet inspection over encrypted traffic. In: Proceedings of the 2015 ACM conference on special interest group on data communication; 2015. p. 213–26.
[208] Khatibi S, Jano A. June. Elastic slice-aware radio resource management with AI-traffic prediction. In: 2019 European conference on networks and communications (EuCNC). IEEE; 2019. p. 575–9.
[209] Khatibi S, Jano A. June. Elastic slice-aware radio resource management with AI-traffic prediction. In: 2019 European conference on networks and communications (EuCNC). IEEE; 2019. p. 575–9.
[210] Homma S, Nishihara H, Miyasaka T, Galis A, Ram OVV, Lopez D, Contreras-Murillo L, Ordonez-Lucena J, Martinez-Julia P, Qiang L, Rokui R, Ciavaglia L, de Foy X. Network slice provision models. Internet Engineering Task Force; 2019. Technical report.
[211] Bega D, Gramaglia M, Garcia-Saavedra A, Fiore M, Banchs A, Costa-Perez X. Network slicing meets artificial intelligence: an AI-based framework for slice management. IEEE Commun Mag 2020;58(6):32–8.
[212] 3GPP. Technical specification group services and system aspects; telecommunication management; study on management and orchestration of network slicing for next generation network. 3rd Generation Partnership Project; 2018. Technical report.
[213] Sun G, Gebrekidan ZT, Boateng GO, Ayepah-Mensah D, Jiang W. Dynamic reservation and deep reinforcement learning based autonomous resource slicing for virtualized radio access networks. Ieee Access 2019;7:45758–72.
[214] MarketsandMarkets. https://www.marketsandmarkets.com/.
[215] Kalashnikov D, Varley J, Chebotar Y, Swanson B, Jonschkowski R, Finn C, Levine S, Hausman K. MT-opt: continuous multi-task robotic reinforcement learning at scale. arXiv preprint, 2104.08212; 2021.
[216] Jiang W, Anton SD, Schotten HD. September. Intelligence slicing: a unified framework to integrate artificial intelligence into 5G networks. In: 2019 12th IFIP wireless and mobile networking conference (WMNC). IEEE; 2019. p. 227–32.
[217] Gutierrez-Estevez DM, et al. Artificial intelligence for elastic management and orchestration of 5G networks. In: IEEE wireless communications, vol. 26; October 2019. p. 134–41. https://doi.org/10.1109/MWC.2019.1800498. no. 5.
[218] Chebotar Y, Hausman K, Lu Y, Xiao T, Kalashnikov D, Varley J, Irpan A, Eysenbach B, Julian R, Finn C, Levine S. Actionable models: unsupervised offline reinforcement learning of robotic skills. arXiv preprint, 2104.07749; 2021.
[219] Peuscher DW. The resource orchestration theory as contributor to supply chain management: an assessment on its applicability. Bachelor's thesis. University of Twente; 2016.
[220] Salhab N, Langar R, Rahim R. 5G network slices resource orchestration using Machine Learning techniques. Comput Network 2021;188:107829.
[221] 3GPP TS 23.501. System Architecture for the 5G system. Rel March 2018;15.
[222] Caballero P, Banchs A, De Veciana G, Costa-Pérez X. Multi-tenant radio access network slicing: statistical multiplexing of spatial loads. IEEE/ACM Trans Netw 2017;25(5):3044–58.
[223] Afolabi I, Taleb T, Samdanis K, Ksentini A, Flinck H. Network slicing and softwarization: a survey on principles, enabling technologies, and solutions. IEEE Commun Surv Tutorials 2018;20(3):2429–53.
[224] Lin M, Zhao Y. Artificial intelligence-empowered resource management for future wireless communications: a survey. In: China communications, vol. 17; March 2020. p. 58–77. https://doi.org/10.23919/JCC.2020.03.006. no. 3.
[225] Lee AX, Devin CM, Zhou Y, Lampe T, Bousmalis K, Springenberg JT, Byravan A, Abdolmaleki A, Gileadi N, Khosid D, Fantacci C. June. Beyond pick-and-place: tackling robotic stacking of diverse shapes. In: 5th annual conference on robot learning; 2021.
[226] DeepMind-Oel AS, Mahajan A, Barros C, Deck C, Bauer J, Sygnowski J, Trebacz M, Jaderberg M, Mathieu M, McAleese N, Bradley-Schmieg N. Open-ended learning leads to generally capable agents. arXiv preprint, 2107.12808; 2021.
[227] Co-Reyes JD, Miao Y, Peng D, Real E, Levine S, Le QV, Lee H, Faust A. Evolving reinforcement learning algorithms. arXiv preprint, 2101.03958; 2021.
[228] Vassilaras S, Gkatzikis L, Liakopoulos N, Stiakogiannakis IN, Qi M, Shi L, Liu L, Debbah M, Paschos GS. The algorithmic aspects of network slicing. IEEE Commun Mag 2017;55(8):112–9.
[229] Li R, Zhao Z, Sun Q, Chih-Lin I, Yang C, Chen X, Zhao M, Zhang H. Deep reinforcement learning for resource management in network slicing. IEEE Access 2018;6:74429–41.
[230] Kibalya G, Serrat J, Gorricho JL, Pasquini R, Yao H, Zhang P. October. A reinforcement learning based approach for 5G network slicing across multiple domains. In: 2019 15th international conference on network and service management (CNSM). IEEE; 2019. p. 1–5.
[232] Taleb T, Afolabi I, Samdanis K, Yousaf FZ. On multi-domain network slicing orchestration architecture and federated resource control. IEEE Network 2019;33(5):242–52.
[233] Guan W, Zhang H, Leung VC. Customized slicing for 6G: enforcing artificial intelligence on resource management. IEEE Network; 2021.
[234] Messaoud S, Bradai A, Ahmed OB, Quang P, Atri M, Hossain MS. Deep federated Q-learning-based network slicing for industrial IoT. IEEE Trans Ind Inf 2020;17(8):5572–82.
[235] Swapna AI, Rosa RV, Rothenberg CE, Pasquini R, Baliosian J. November. Policy controlled multi-domain cloud-network slice orchestration strategy based on reinforcement learning. In: 2020 IEEE conference on network function virtualization and software defined networks (NFV-SDN). IEEE; 2020. p. 167–73.
[236] Liu Q, Han T, Zhang N, Wang Y. DeepSlicing: deep reinforcement learning assisted resource allocation for network slicing. arXiv preprint, 2008.07614; 2020.
[237] Xi R, Chen X, Chen Y, Li Z. December. Real-time resource slicing for 5G RAN via deep reinforcement learning. In: 2019 IEEE 25th international conference on parallel and distributed systems (ICPADS). IEEE; 2019. p. 625–32.
[238] Ginige NU, Manosha KS, Rajatheva N, Latva-aho M. May. Admission control in 5G networks for the coexistence of eMBB-URLLC users. In: 2020 IEEE 91st vehicular technology conference (VTC2020-Spring. IEEE; 2020. p. 1–6.
[239] Ojijo MO, Falowo OE. A survey on slice admission control strategies and optimization schemes in 5G network. IEEE Access 2020;8:14977–90.
[240] Han B, Feng D, Schotten HD. A Markov model of slice admission control. IEEE Network Lett 2018;1(1):2–5.
[241] Han B, DeDomenico A, Dandachi G, Drosou A, Tzovaras D, Querio R, Moggio F, Bulakci O, Schotten HD. October. Admission and congestion control for 5g network slicing. In: 2018 IEEE conference on standards for communications and networking (CSCN). IEEE; 2018. p. 1–6.
[242] Han B, Sciancalepore V, Feng D, Costa-Perez X, Schotten HD. April. A utility-driven multi-queue admission control solution for network slicing. In: IEEE INFOCOM 2019-IEEE conference on computer communications. IEEE; 2019. p. 55–63.
[243] Challa R, Zalyubovskiy VV, Raza SM, Choo H, De A. Network slice admission model: tradeoff between monetization and rejections. IEEE Syst J 2019;14(1):657–60.
[244] Fu Y, Wang S, Wang CX, Hong X, McLaughlin S. Artificial intelligence to manage network traffic of 5G wireless networks. IEEE Network 2018;32(6):58–64.

[245] Peterson L, Anderson T, Culler D, Roscoe T. A blueprint for introducing disruptive technology into the internet. Comput Commun Rev 2003;33(1):59–64.
[247] aaijmakers Y, Mandelli S, Doll M. Reinforcement learning for admission control in 5G wireless networks. arXiv preprint, 2104.10761; 2021.
[248] Han B, Tayade S, Schotten HD. July. Modeling profit of sliced 5G networks for advanced network resource management and slice implementation. In: 2017 IEEE symposium on computers and communications (ISCC). IEEE; 2017. p. 576–81.
[249] Bega D, Gramaglia M, Banchs A, Sciancalepore V, Samdanis K, Costa-Perez X. May. Optimising 5G infrastructure markets: the business of network slicing. In: IEEE INFOCOM 2017-IEEE conference on computer communications. IEEE; 2017. p. 1–9.
[251] Khodapanah B, Awada A, Viering I, Barreto AN, Simsek M, Fettweis G. Slice management in radio access network via deep reinforcement learning. 2020, May.
[252] Jiang C, Zhang H, Ren Y, Han Z, Chen KC, Hanzo L. Machine learning paradigms for next-generation wireless networks. IEEE Wireless Commun 2016;24(2):98–105.
[253] Wang T, Vandendorpe L. June. Successive convex approximation based methods for dynamic spectrum management. In: 2012 IEEE international conference on communications (ICC). IEEE; 2012. p. 4061–5.
[254] Ouyang H, He N, Tran L, Gray A. February. Stochastic alternating direction method of multipliers. In: International conference on machine learning. PMLR; 2013. p. 80–8.
[256] Bega D, Gramaglia M, Banchs A, Sciancalepore V, Costa-Pérez X. A machine learning approach to 5G infrastructure market optimization. IEEE Trans Mobile Comput 2019;19(3):498–512.
[257] Zanzi L, Sciancalepore V, Garcia-Saavedra A, Costa-Pérez X. April. OVNES: demonstrating 5G network slicing overbooking on real deployments. In: IEEE INFOCOM 2018-IEEE conference on computer communications workshops (INFOCOM WKSHPS). IEEE; 2018. p. 1–2.
[258] Yan M, Feng G, Zhou J, Sun Y, Liang YC. Intelligent resource scheduling for 5G radio access network slicing. IEEE Trans Veh Technol 2019;68(8):7691–703.
[259] Li R, Zhao Z, Sun Q, Chih-Lin I, Yang C, Chen X, Zhao M, Zhang H. Deep reinforcement learning for resource management in network slicing. IEEE Access 2018;6:74429–41.
[260] Bakri S, Brik B, Ksentini A. On using reinforcement learning for network slice admission control in 5G: offline vs. online. Int J Commun Syst 2021;34(7):e4757.
[261] ETSI. Next generation protocols (ngp); e2e network slicing reference framework and information model. European Telecommunications Standards Institute; 2018. Technical report.
[262] Raza MR, Natalino C, Öhlen P, Wosinska L, Monti P. Reinforcement learning for slicing in a 5G flexible RAN. J Lightwave Technol 2019;37(20):5161–9.
[263] Bouzidi EH, Outtagarts A, Langar R. December. Deep reinforcement learning application for network latency management in software defined networks. In: 2019 IEEE global communications conference (GLOBECOM). IEEE; 2019. p. 1–6.
[264] Vincenzi M, Lopez-Aguilera E, Garcia-Villegas E. Maximizing infrastructure providers' revenue through network slicing in 5G. IEEE Access 2019;7:128283–97.