IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 24 March 2023, accepted 13 April 2023, date of publication 17 April 2023, date of current version 25 April 2023.
Digital Object Identifier 10.1109/ACCESS.2023.3267985

SURVEY

Machine Learning in Network Slicing–A Survey

HNIN PANN PHYU[1], (Student Member, IEEE), DIALA NABOULSI[1], (Member, IEEE), AND RAZVAN STANICA[2]
[1]Département de Génie Logiciel et des Technologies de l'information, École de Technologie Supérieure, Université du Québec, Montreal, QC H3C 1K3, Canada
[2]Inria, CITI, INSA Lyon, University of Lyon, 69100 Villeurbanne, France

Corresponding author: Hnin Pann Phyu (hnin.pann-phyu.1@ens.etsmtl.ca)

This work was supported in part by the National Natural Sciences and Engineering Research Council of Canada (NSERC) under Grant RGPIN-2020-06050, and in part by the Fonds de Recherche du Québec—Nature et Technologies (FRQNT) under Grant FRQNT 2022-NC-297403.

ABSTRACT 5G and beyond networks are expected to support a wide range of services, with highly diverse requirements. Yet, the traditional ''one-size-fits-all'' network architecture lacks the flexibility to accommodate these services. In this respect, network slicing has been introduced as a promising paradigm for 5G and beyond networks, supporting not only traditional mobile services, but also vertical industries services, with very heterogeneous requirements. Along with its benefits, the practical implementation of network slicing brings a lot of challenges. Thanks to the recent advances in machine learning (ML), some of these challenges have been addressed. In particular, the application of ML approaches is enabling the autonomous management of resources in the network slicing paradigm. Accordingly, this paper presents a comprehensive survey on contributions on ML in network slicing, identifying major categories and sub-categories in the literature. Lessons learned are also presented and open research challenges are discussed, together with potential solutions.

INDEX TERMS Network slicing, 5G network, machine learning.

I. INTRODUCTION

The number of fifth generation (5G) mobile networks subscribers is forecast to reach 3.5 billion globally by 2026 [1]. The average data usage is estimated to reach 35 GB/month/user, resulting from 400 5G use cases in 70 industries [2]. Indeed, 5G is expected to play a major role in the digitalization of various vertical markets, such as automotive, smart grid and the Internet of Things (IoT). A wide range of use cases with highly diverse requirements are envisioned to be supported [3]. These use cases can be roughly grouped into three categories: Extreme Mobile Broadband (xMBB), Ultra-Reliable and Low-Latency Communications (URLCC), and Massive Machine Type Communication (mMTC) applications [4].

Previous generations of mobile networks, i.e. 2G, 3G and 4G, were designed to efficiently handle human-type communication. However, their ''one-size-fits-all'' architectures lack the flexibility required to accommodate the diverse requirements of future 5G and beyond use cases [5]. As a result, the concept of network slicing has recently been introduced by the Next Generation Mobile Network (NGMN) alliance [6] to allow mobile network operators to support this increasing variety of use cases. Network slicing consists of creating multiple logical networks on top of a single physical network, on a per-service basis [7]. Thus, these logical networks (i.e. network slices) can be formed and customized to different scenario requirements in terms of functionality, performance and isolation, as underlined by the 3rd Generation Partnership Project (3GPP) [8]. From a business point of view, it is estimated that 30% of 5G operators revenue will be driven by network slicing [9].

Enabling the vision of network slicing cannot be achieved without fully automating overall network operations. In recent years, significant effort has been put in this direction, through the integration of machine learning (ML) techniques [19]. Traffic flows generated from 5G services are increasingly heterogeneous and exhibit complex

The associate editor coordinating the review of this manuscript and approving it for publication was Alessandro Floris.

TABLE 1. Comparisons of existing surveys on network slicing.

Ref	Resource Management Model	Supervised/Unsupervised	Reinforcement Learning	Scope
[3]	●	○	○	5G network slicing architectures
[10]	○	○	○	Network slice creation models and slicing templates proposed by SDOs
[11]	○	○	○	5G network slicing development and its integration with the MEC and the cloud
[12]	●	○	○	Optimization frameworks for network slicing
[13]	◕	○	◑	Algorithmic issues for admission control and resource allocation aspects in network slicing, including RL methods
[14]	◑	◔	◔	Importance of network slicing and ML in 5G-enabled IoT applications
[15]	◑	◔	◔	5G network slicing development requirements for IoT applications
[16]	●	◔	○	Mathematical modelling encompassing game theory models, prediction models, failure recovery models in resource allocation methods
[17]	◕	○	●	Admission control, resource allocation and resource orchestration aspects in network slicing with DRL approaches
[18]	●	◑	◑	DRL-based contributions in network slicing
Our Survey	●	●	●	ML-based algorithmic approaches in network slicing

correlations [20]. In this case, it is not possible to rely on conventional mathematical models and algorithms to process them [21]. Conversely, recent advancements in ML techniques, with their ability to process large amounts of data and their efficiency in unveiling complex correlations in datasets, are positioning ML techniques as very promising solutions in the automation of network slicing operations.

Accordingly, a large amount of studies has been conducted, introducing ML-driven algorithmic solutions in the context of network slicing. The number of publications on the topic keeps growing significantly over the years, with 340 papers in 2022 alone, as shown in Figure 1. In the light of 6G standardisation and the continuous development of ML techniques, a higher interest in the topic is further expected in the coming years. Considering the most recent and noticeable works, this survey thus aims at providing a comprehensive review of ML solutions in network slicing. In addition, it underlines clear research directions for those who wish to further investigate ML techniques to address the complex problems in network slicing, for existing and future mobile networks.

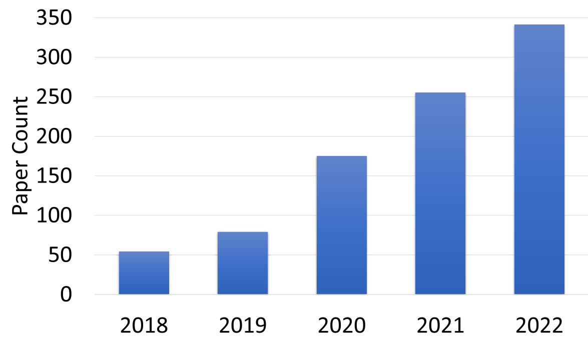


FIGURE 1. Number of yearly publications on machine learning in network slicing (source: Google scholar).

A. SCOPE OF THE SURVEY

Several surveys have already been published on the subject of network slicing. A few of them review network slicing architectures and principles [3], [10], [11], while others focus on the algorithmic aspects of network slicing [12], [13] or on its mathematical modeling [16]. Some surveys [14], [15] discuss both architecture and algorithmic aspects in network slicing. Two recent surveys focus on very specific applications of deep reinforcement learning (DRL) in a network slicing context [17], [18].

More precisely, in [3], the authors present the architecture of 5G sliced networks, with its different layers: infrastructure layer, network function layer, management and orchestration (MANO), and service layer. The authors in [10] investigate the network slices ordering and creation models proposed by different standard developing organizations (SDOs). They analyze the key attributes and functions of the most common models and propose unified network slicing models for efficient end-to-end (E2E) network slicing management. Furthermore, the authors of [11] discuss the potential and integration of multi-access edge computing (MEC) and cloud technologies in network slicing. However, algorithmic aspects of resource management perspectives in network slicing have not been studied in these surveys.

In [12], the authors review MANO methods and algorithmic approaches in network slicing. However, their focus is only on the optimization frameworks and operational research methodologies. Accordingly, their survey does not include machine learning-based approaches. In the study in [13], the authors emphasize the admission control and resource allocation aspects of network slicing, and recent approaches for these problems, including reinforcement learning (RL), are discussed and documented. Nevertheless, no other ML methods (i.e. supervised and unsupervised approaches) are analyzed. The works in [14] and [15] emphasize the role of 5G network slicing in IoT applications. However, algorithmic aspects, in particular advaned ML-based solutions, are not well-grounded.

In [16], the discussed mathematical models encompass game theory models, prediction models, and failure recovery models. Relationships among them are analyzed in the realm of resource allocation for network slicing across multiple domains. However, only supervised learning approaches are discussed, and only in the context of prediction models.

Closer to our scope, the work in [17] goes over a DRL driven network slicing resource management model, discussing the objective, the network domains, the Markov decision process (MDP) modelling and the use cases. However, forecasting related problems are not covered by this study. Furthermore, the authors in [18] survey a similar topic: the feasibility of DRL frameworks in the 5G network slicing paradigm. However, these studies dedicate little to no discussion to other ML techniques, like the role of supervised and unsupervised learning in the traffic forecasting function, or to multi-armed bandit techniques which have been widely used in the network slicing resource management regime. Besides, the topics of admission control (either inter-slice or intra-slice) and resource allocation granularity (i.e. coarse-grained or fine-grained) are not well-elaborated in these prior works.

To summarize, none of the contemporary surveys provides a comprehensive review of ML-based approaches in network slicing, with the relevant background information. Specifically, none of the existing surveys articulated thoroughly the various business models in network slicing and projected them with respect to the network slicing architecture, as we do below. This allows for a clear mapping of ML-based approaches to the different business entities. Moreover, unlike other works (i.e. [17], [18]), our survey proposes an original taxonomy regarding the granularity level of resource management solutions in the context of network slicing. On the other hand, we cover at an unprecedented level of detail the resource management models used in network slicing and the application of ML-based methods to this end. To say the least, we extensively cover solutions with unsupervised and supervised ML techniques, as well as RL techniques, which is unique in the field. Besides, not only do we extensively identify the spectrum of research gaps, but also we put forward the potential ML-based solutions respectively. The summary of the scope of existing surveys compared to ours is indicated in Table 1.

B. SURVEY ORGANIZATION

The remainder of this survey paper is organized as follows: Section II equips the reader with necessary background information on network slicing. In Section III, the functioning of ML techniques commonly used in the context of network slicing is explained. We remark that readers with strong ML background can completely skip Section III. Section IV reviews ML contributions in network slicing. Identified open challenges and opportunities can be found in Section V. Finally, we conclude the survey in Section VI. The detailed structure of the survey is further depicted in Figure 2.

II. NETWORK SLICING

A. NETWORK SLICING CONCEPT

The concept of network slicing has been introduced by the NGMN alliance [6] as a solution for accommodating the diverse requirements of 5G and beyond use cases. Technically, network slicing consists of creating logical networks on top of a single physical network, across multiple domains, on a per-service basis. The resulting network slices could be managed independently, mutually isolated, and created on-demand [6].

End-to-end network slicing refers to creating network slices that cover the entire communication path, from the radio access network (RAN) to the core network (CN). As network slicing is leveraged based on a virtualized infrastructure powered by network function virtualization (NFV) and software defined networks (SDN), it enables flexible and programmable control of network resources while respecting service level agreements (SLA) [23]. Undoubtedly, these logical networks (i.e. network slices) could be a game-changer for potential 5G use cases as they can be established and customized to different use case requirements in terms of functionality, performance, and isolation, as emphasized by the Third Generation Partnership Project (3GPP) [24]. Figure 3 formally illustrates the network slicing architecture as defined by NGMN [22]. As shown in the figure, the architecture contains the following three layers:

- **Application Layer:** It consists of end-user services (e.g., smart home/city, remote surgery, and ultra High Definition (HD) video streaming). A Service Instance has a specific type (i.e. eMBB, URLCC, or mMTC).
- **Network Layer:** It consists of network slice instances, including logical and physical resources. A network slice instance supports one or more service instances and is composed of one or more sub-network instances. For brevity, we use in the following the word "slice" to refer to a network slice instance.
- **Resource Layer:** It includes shared infrastructure resources with both physical and virtual resources that are controlled by the NFV/SDN framework. The resource layer provides all the required resources to the network slice instance layer.

B. BUSINESS MODELS IN NETWORK SLICING

The business model adopted in most of the existing research works on network slicing includes three business entities: application service provider (ASP), mobile virtual

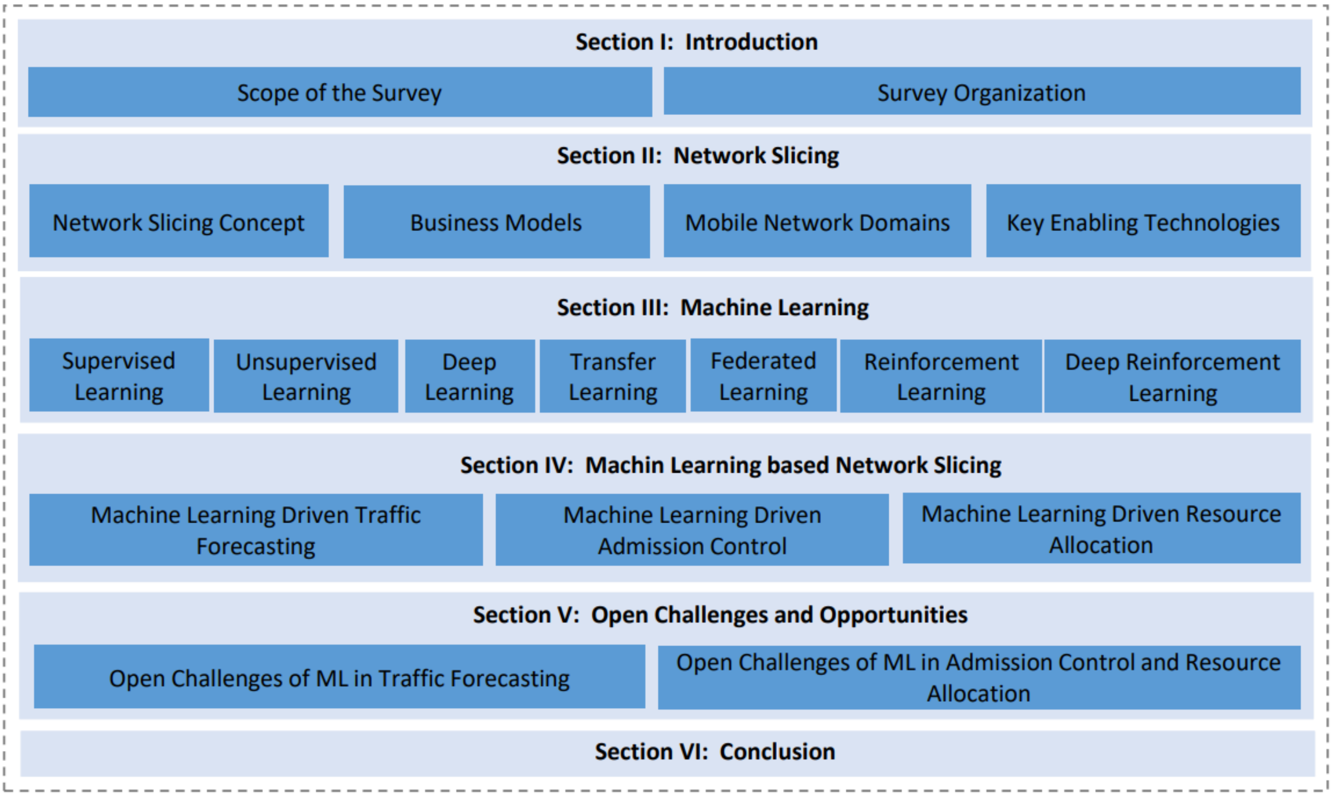


FIGURE 2. Structure of the survey.

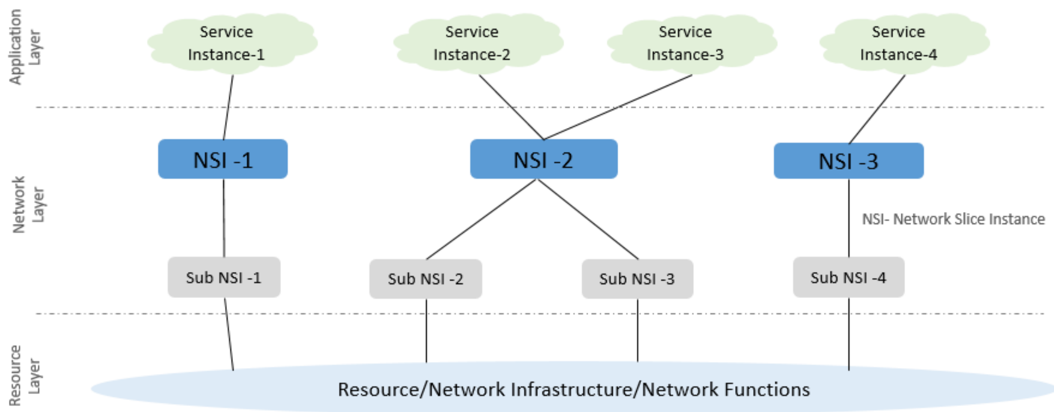


FIGURE 3. NGMN network slicing concept [22].

network operator (MVNO) and infrastructure provider (InP) [25], [26]. An ASP offers a service to end-users by using a slice operated by an MVNO. For this purpose, the ASP provides the service quality of service (QoS) requirements to the MVNO. It also pays the MVNO the cost associated to offering the service, based on the slice instance running time, the number of served customers, and the coverage area. The MVNO creates the slice based on the received service requirements. It requests as well from the InP to allocate physical and virtual resources for implementing the slice. The MVNO continuously monitors the QoS of each slice instance to ensure requirements are met. One or more slice instances may belong to one MVNO. The InP owns the physical substrate network infrastructure, manages the life-cycle of physical and virtual resources and provides a complete set of resources for slices. It is important to stress that it is common in the literature to have ASP functionalities covered by the MVNO, especially in the network slicing resource allocation problem formulation (see for instance [27], [28], [29], [30], [31], [32], [33], [34] to name just a few). Figure 4 summarizes the interactions among business entities according to this business model.

Apart from the previously described business model, more complex ones, with more business entities, were introduced in [35]: *i)* Single-domain business model, with a single infrastructure and a single slice provider; *ii)* Multi-domain business model, with multiple infrastructure providers and multiple slice providers. Figure 5 depicts these two business models. For brevity, we only describe the main entities in these models, also used in several other works (see for instance [27], [36], [37], [38], [39], [40]): slice provider (SIP), slice tenant (ST), slice customer (SC) and service provider (SP).

- **Slice Provider:** It creates a slice based on the slice template endorsed by a slice tenant or a slice customer. Generally, it obtains the resources required for the slice from the InP. However, in some cases [36], the Slice Provider controls directly the infrastructure resources and its ultimate goal is to efficiently allocate resources among multiple Slice Tenants.
- **Slice Tenant:** It requests a slice from the Slice Provider, according to the received demands from the Slice Customers. It also operates the slice. Usually, multiple Slice Tenants rely on the same InP [27], according to different SLAs [40].
- **Slice Customer:** It represents an end user, who can subscribe to one or more slices simultaneously, possibly managed by different Slice Tenants [36]. A Slice Customer can take the role of a Slice Tenant and, by that, serve other Slice Customers [35].
- **Service Provider:** It provides various kinds of slice services to Slice Customers (i.e. end users). In some cases [37], [39], a Service Provider plays the role of a Slice Tenant as well, reserving resources from the InP and offering services to the Slice Customers. In other cases [38], the Service Provider also has control over the infrastructure resources and fulfills slice requests of Slice Customers.

In a nutshell, the business models applied in the surveyed studies can be mapped partially or fully into one of the above-described models. Indeed, a well-defined business model is a critical component of a research work on network slicing. In this context, this is especially true for resource allocation problems, where multiple entities from those listed above are involved.

C. MOBILE NETWORK DOMAINS AND NETWORK SLICING

End-to-end network slices encompass the three domains of RAN, CN, and transport network (TN), possibly from multiple operators [41]. Network slicing can be deployed on a traditional network consisting of these three domains, as the concept is defined on a logical level in the 3GPP functions. However, to fully leverage the slicing concept, these domains have to be integrated with NFV, SDN, MEC, and cloud computing [11], the key drivers of network slicing, described in subsection II-D. In fact, one requirement in network slicing is performance isolation among network slices [3]. Traditional networks fail to meet this requirement. For instance, a similar concept to network slicing is that of data radio bearers (DRBs) in traditional RAN. It allows to handle traffic with different QoS requirements [42]. Yet, DRBs of different users are controlled by a shared medium access control (MAC) protocol and thus do not guarantee performance isolation [43].

In the realm of 5G network slicing, the technical specifications of those domains are being developed in different SDOs: RAN and CN domains are regulated by the 3GPP [44], [45], while the functioning of the TN domain is specified by the Internet Engineering Task Force (IETF) [46] and the Broadband Forum (BBF) [47]. Figure 6 illustrates the high-level architecture of a 5G sliced network.

1) RAN SLICING

The main question in the RAN domain is how to appropriately divide the overall radio spectrum resources for different applications to guarantee the rigid QoS requirements expressed by some network slices [48]. The degree of complexity is higher in RAN than in CN and TN, due to the difficulties in the segregation of radio resources. Moreover, virtualization in the RAN is still in its infancy, unlike at the CN level [49]. In this respect, in the 3GPP 5G RAN specifications, eight possible RAN virtual functions splitting options are considered, based on the Cloud-RAN (C-RAN) concept [50], where RAN functions are split between Remote Radio Units (RRUs) and Baseband Units (BBUs) hosted in the Next Generation Node B (gNB) and a BBU pool, respectively. The RAN functional split allows network slices to share certain RAN functions among each other [51].

With these different functional splitting options, it is important to consider their ability in guaranteeing performance isolation among slices [52]. In fact, RAN slicing done at the spectrum level provides the highest degrees of isolation and customization compared to other RAN slicing options at the Radio Resource Management (RRM) level (i.e. inter-cell interference coordination (ICIC) level, packet scheduling (PS) level, admission control (AC) level) [53]. More specifically, spectrum resources are organized as carriers (i.e. each carrier is composed of resource blocks) in RAN slicing at the spectrum planning level. Each RAN slice tenant is assigned a separate carrier so as to ensure complete performance isolation among slices, thereby enabling the customization of the slices based on the tenant-specific requirements at all levels of the RRM functionalities.

Notably, many researchers are working on the RAN slicing system architecture to build a service-oriented network. More specifically, SoftRAN [54], FlexRAN [55] and Orion [49] solutions present new RAN virtualization models that perform the abstraction of underlying physical resources and ensure efficient resource utilization. Furthermore, these SDN-based RAN architectures decouple the data plane from the control plane and allow slice customers to have full control of their own RAN functionalities and yet guarantee performance isolation [51].

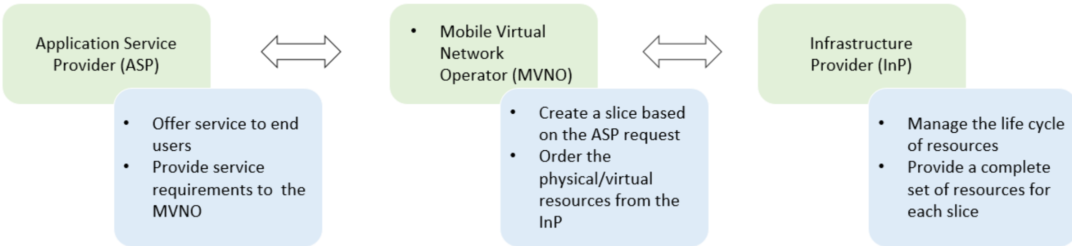
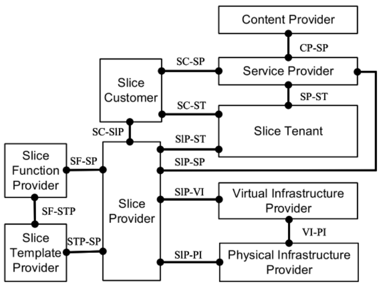
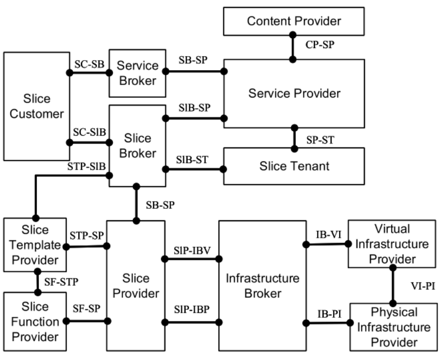


FIGURE 4. Business model of network slicing.



(a) Single-domain business model.



(b) Multi-domain business model.

FIGURE 5. Single-domain and multi-domain business models of network slicing [35].

2) CN SLICING

Recent years have witnessed the transformation of the entire CN network with the help of NFV, SDN, and cloud computing. As 5G is expected to support a variety of use cases with diverse requirements, the 5G core (5GC) will adopt a service-based architecture (SBA), as specified by the 3GPP [56], to enable multiple virtual networks to run on the same physical infrastructure [57]. Moreover, with SDN decoupling the control plane from the user plane and with NFV virtualizing physical network resources, these two technologies bring programmability and flexibility to the deployment, control and management of CN functions [23].

In fact, control and user plane separation (CUPS) is a necessity in 5GC to fulfill the automation requirement of future network operations. Moreover, specific functions are required in 5GC to enable network slicing. Particularly, the network slice selection function (NSSF), introduced under the vision of SBA [56], is responsible for selecting the appropriate set of network slice instances (NSIs) and determining the access and mobility management function (AMF) set to serve UEs. Technically, a UE can be served by a maximum of up to eight network slices simultaneously [58]. In this case, AMF is in charge of the UE association with corresponding slices [59]. If some CN functions, like AMF and NSSF, can be shared by multiple NSIs, others, i.e. user plane function (UPF) or unified data management (UDM), are slice-specific [60]. For brevity, we do not describe all the new 5GC functions in this survey and interested users are referred to [45] for more information.

Finally, for a fully-fledged network slicing system, 5GC needs to adopt a cloud-native based design. By that, conventional VNFs, deployed traditionally on virtual machines (VMs), are transformed into cloud-native network functions (CNFs) that are deployed on containers instead [11]. Specifically, running CNFs on containers facilitates automation in the premises of a cloud environment [61].

3) TN SLICING

The TN domain is as important as the RAN and CN domains to leverage the benefits of network slicing [62]. With 5G, the TN slices are expected to carry the exponentially-increasing traffic load and satisfy stringent SLAs [63]. According to the IETF, a TN slice is a logical network topology connecting various endpoints in the RAN and the CN, with appropriate shared or dedicated network links, that are used to ensure specific SLAs [46].

TN links can be established using different existing technologies (e.g., optical fiber, Ethernet, microwave) [10]. In fact, Multi-Protocol Label Switching (MPLS) in TN provides the adaptation of different TN layer technologies, thereby enabling the multi-service mobile transmission [52]. In addition, already developed technologies, such as Flexible Ethernet (FlexE), Wavelength Division Multiplexing (WDM), and Optical Transport Network (OTN), can be used to ensure the performance isolation among slices [64]. Ongoing research efforts are aiming at evolving these solutions for the purpose of TN slicing. Particularly, elastic optical networks (EONs) [65], Optical Virtual Networks (OVN) [66] and Open Optical Network (OON) [67] embrace service-oriented TN slicing by providing scalability, flexibility and inter-operability on optical TNs.

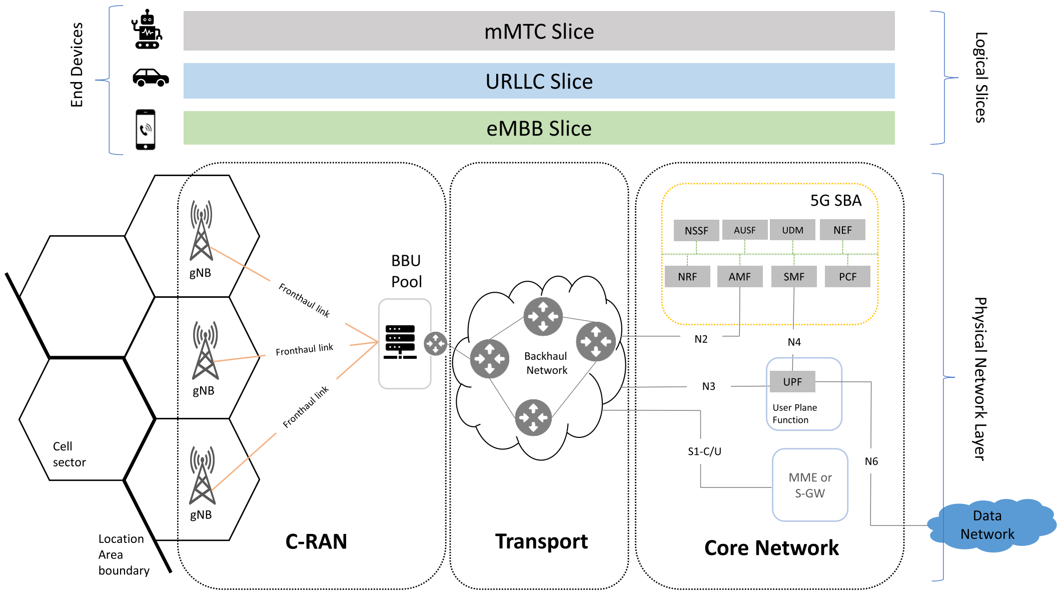


FIGURE 6. High-level architecture of end-to-end network slicing.

Besides, SDN-driven TN represent a very prominent solution to leverage the development of cloud-based services in 5G systems [68]. In fact, multiple TN slices are required for the E2E slice provisioning with different RAN deployment scenarios (i.e. distributed RAN, centralized RAN and cloud RAN) [46]. Hence, a unified service orchestrator is required to control multiple TN slices from multiple network domains and to integrate fronthaul and backhaul networks [69]. To cope with this, the IETF establishes a management and control framework of TN slices, under the name of Abstracting and Control of Traffic engineer Network (ACTN), to allow an MVNO to manage multiple network domains with a single abstract network [70].

D. KEY ENABLING TECHNOLOGIES OF NETWORK SLICING

Virtualization technologies have brought up enormous advantages in terms of programmability and flexibility for resource allocation in end-to-end network slicing. Specifically, NFV, SDN, MEC and cloud computing are the major catalysts to facilitate network slicing. Hereafter, the role of each of these enabling technologies in network slicing is discussed.

1) NETWORK FUNCTION VIRTUALIZATION

NFV decouples the network functions from their proprietary hardware and runs them as software on general purpose servers. The architectural framework of NFV is introduced by the European Telecommunications Standards Institute (ETSI) [71]. It is composed of VNFs, NFV infrastructure (NFVI), and NFV management and orchestration (MANO). VNFs are the virtualized network elements that can be chained together in a particular order to form service function chains (SFC) offering one specific service [72]. A network slice for one specific service is commonly represented as one SFC. NFVI encompasses both physical and virtual resources where VNFs are deployed. Figure 7 depicts the architecture of ETSI NFV MANO that enables the automation of resource management, network services, and VNFs to guarantee the network performance requirements of operators. This architecture consists of three main functional blocks: the NFV orchestrator (NFVO), the VNF manager (VNFM), and the virtualized infrastructure manager (VIM). The NFVO orchestrates the NFVI resources and manages the life cycle of network services. The VNFM is in control of the instantiating, monitoring, and termination of the VNF instances. Each VNF instance is controlled by a VNFM. Finally, the VIM is responsible for governing the computation, storage, and networking resources. Those functional blocks not only communicate to other function blocks of the NFV architecture, but also interact among each other through a set of reference points [8].

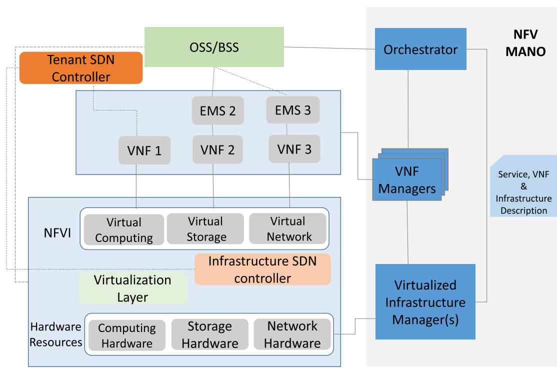


FIGURE 7. An integration of SDN controllers into the ETSI NFV reference architecture at the two levels required to achieve network slicing (inspired from [23]).

2) SOFTWARE DEFINED NETWORKING

SDN decouples the control plane from the data plane and places it on a logically centralized controller. SDN is one of the key enablers of network slicing, as it enables programmability, flexibility, service-oriented adaptation, scalability, and robustness [73]. Besides, SDN allows the MVNOs to manage and control their allocated resources through the abstract view of the network. The SDN controller manages network slices effectively by applying rules when necessary and in accordance with the corresponding network policy. The convergence of SDN and NFV is a commonly used mechanism in the deployment of E2E network services. As shown in Figure 7, ETSI proposed an architecture where NFV MANO is integrated with two SDN controllers: *i)* infrastructure SDN controller (ISDNC) to provide the required connectivity among VNFs and its components by managing the underlying network resources and, *ii)* tenant SDN controller (TSDNC) to manage dynamically the corresponding VNFs. While the TSDNC provides an overlay comprising tenant VNFs that define the network service(s), the ISDNC provides an underlay to support the deployment and connectivity of VNFs [23].

3) MOBILE EDGE AND CLOUD COMPUTING

MEC and cloud computing offer on-demand storage, computational and networking resources within a single or multiple platforms [74]. The appealing idea of MEC for network slicing is to bring the network functions and related applications closer to the end-users, to reduce delays and burdens on the back-haul. Simply speaking, in the realm of 5G, virtualized resources of RAN (i.e. BBU and RRU) and some user-plane functions of the CN could be located in the edge cloud to provide low-latency services (i.e. URLCC). With the help of SDN, 5G networks are able to control the VMs distributed in the cloud core and edge cloud in a centralized manner.

III. MACHINE LEARNING

Most of the network slicing optimization problems have been formulated as either Mixed-integer Linear Programming (MILP) or Nonlinear Integer Programming (NLP) problems. Generally, the resulting formulations are proven to be NP-Hard. Thus, their global optimal solutions cannot be obtained within a polynomial time, especially for large-scale instances. To deal with this, the typical approach is to decompose the complex optimization problem into simpler sub-problems or relax complicating constraints of the problem and solve a simpler version. Then, low computational-complexity algorithms (i.e. heuristics, metaheuristics, genetic algorithms (GA), and game theory) are applied to derive a solution close to the optimal one.

Regardless of being less computational expensive, the major shortcoming of traditional optimization models and conventional algorithms is that they lack the flexibility to adapt to today's highly dynamic and fast-changing network environment with 5G heterogeneous services and massive connections [75]. Therefore, algorithms with high adaptability are needed to make real-time decisions in network slicing.

When it comes to traffic forecasting, conventional solutions like ARIMA [76] and Holt-Winters [77] have been widely used over the years. Compared with those solutions that could not extract and predict the complicated spatiotemporal features of mobile traffic in presence of user mobility [78], solid forecasting performance in this field has been demonstrated in the recent years by machine learning based forecasting solutions, deep learning based techniques in particular.

Needless to say, ML-based algorithms are thus envisioned as a promising solution in the realm of 5G network slicing problems [79]. That being said, before delving deeper in the machine learning techniques applied to network slicing resource control and management problems, we briefly summarise them in Table 2.

IV. MACHINE LEARNING BASED NETWORK SLICING

Works applying ML techniques in the context of network slicing can be grouped into three categories: *i)* traffic forecasting, *ii)* admission control, and *iii)* resource allocation. These categories reflect three key network slicing building blocks that together aim at ensuring network slicing SLAs are respected. Figure 8 illustrates the relationships among these building blocks, as commonly adopted in resource management models in the literature (*e.g.*, [77], [80]).

The traffic forecasting block allows to predict the evolution of traffic load and resource usage for slices, over future time instants. The outcome of the traffic forecasting solution can be fed into the slice admission control solution and into the slice resource allocation solution to enable better decisions (*e.g.*, maximize system resource utilization).

The admission control block decides on the slices/users to be served in the future, according to various aspects (*e.g.*, resource availability, resource efficiency or operator revenue [77]). It can also build on the outcome of the traffic forecasting block for refining admission decisions in an anticipatory way. Once a slice/user is admitted, the resource allocation block assigns the resources to each slice/user by avoiding the over-provisioning and under-provisioning of the resources and ensuring the SLAs are respected [81]. Hereafter, we survey existing ML-based contributions in network slicing under the aforementioned three categories.

A. TRAFFIC FORECASTING

It is beneficial to know the required resources per slice in advance, over a certain time interval. This allows to pre-assign the resources to the slices and avoid SLAs violations [82]. Over the years, non-ML solutions like ARIMA [76] and Holt-Winters [77] have been widely used for temporal forecasting, although they are not suitable to extract and predict the complicated spatiotemporal features of mobile traffic in presence of user mobility [78]. Because of this, many alternative ML-based forecasting techniques,

TABLE 2. Summary of common ML techniques in network slicing.

ML Technique	Learning Approache	Role in network slicing	Description
Random Forest	Supervised	Classification and regression of network traffic and QoS KPIs	Decision trees are used to infer the model and derive decisions based on the majority of the votes from all decisions trees.
K-Nearest Neighbors (K-NN)		Classification and regression of network traffic and QoS KPIs	k-closest labeled data points are identified using Euclidean or Manhanttan distance metrics and a decision is made based on how close two data points are to each other.
Gradient Boosting		Classification and regression of network traffic and QoS KPIs	The outputs of different decision trees are combined to generate decision. A gradient descent procedure is used to minimize the loss when combining the decision trees.
k-means	Unsupervised	Clustering of end-users or network nodes	Group a set of data points into a predefined number of clusters k. It then assigns the data points to their respective closest clusters.
Autoencoder		Network slicing data dimension reduction	Feed-forward neural network is used to learn a representation for un-labelled data. It consists of three components: encoder (i.e. input layer), code (i.e. hidden layer) and decoder (i.e. output layer).
Generative Adversarial Network (GAN)		Generate data with similar characteristics to the real one	Two neural networks are used in this architecture: the generator and the discriminator. The generator network creates fake data samples, while considering feedback from the discriminator. The discriminator network classifies fake samples as either real or fake.
Artificial Neural Network (ANN)	Supervised/unsupervised	Extract complex knowledge from a representative network dataset	It is composed of three layers: input, hidden and output. It is also a feed-forward neural network because data flows from the input layer to the output layer only in forward direction, without going backward.
Deep Neural Network (DNN)		Assist DRL algorithms as a function approximator in network slicing resource control and management	A DNN is an ANN with more than three hidden layers.
Recurrent Neural Network (RNN)		Sequential data modeling, traffic forecasting and capturing the temporal variations of service requests in user mobility	Feedback loops in RNN allow information to flow back into the previous parts of the neural network, giving them the capability of processing and capturing sequences in the input data. Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) networks are special kinds of RNNs.
Convolutional Neural Network (CNN)		Spatial data modeling, traffic forecasting and classification in network slicing problems	The first layer in the CNN network is a convolution layer. It is followed by other convolution layers or max-pooling layers. Each convolution layer is interleaved by a dropout layer which is used to reduce the over-fitting. Finally, Fully Connected (FC) layers are used to generate the final desired output.
Transfer Learning (TL)		Extract knowledge from a previous network environment	TL allows to benefit from a pre-trained ML model to solve a new problem or solve the same problem in a new environment, as long as similar contexts are considered. Technically, it transfers as much relevant knowledge as possible from a source pre-trained model to a target model.
Federated Learning (FL)	Supervised/unsupervised/ reinforcement learning	Decentralized multi-agents collaborative learning and privacy preserving for traffic forecasting, resource control and management	FL allows to train the local models and store the corresponding datasets locally, without exchanging them. However, local models can gain knowledge from each other via shared global model parameters.
Q-Learning	Reinforcement Learning	Decision making for network slicing resource control and management	Q-learning is a value-based learning and relies on a Q-table that reflects the Q-value (quality) of each action in a given state. The ultimate goal of Q-Learning is to find the maximum Q-table.
epsilon-greedy		Decision making for network slicing resource control and management	Selects an action using the greedy policy with a probability of $1-\varepsilon$ and a random action with a probability of ε.
Upper confidence bounds (UCB)		Decision making for network slicing resource control and management	Select an action with high potential to get an optimal value, by considering an upper confidence bound on the reward value.
Thompson Sampling		Decision making for network slicing resource control and management	A probabilistic approach, with an action selected based on its probability of being optimal. It is also called posterior sampling algorithm as its rewards are estimated from the posterior reward distribution.
Deep Q Network (DQN)	Deep Reinforcement Learning	Decision making for network slicing resource control and management with high dimensional inputs	DQN uses two Q-networks: current Q-network is used to select the action and target Q-Network is used to take the values of the corresponding state-action value. DNN is used to estimate the Q-value function.
Dueling Deep Q Network (DDQN)		Decision making for network slicing resource control and management with high dimensional inputs	DDQN has a single Q-network with a two-stream Q-function: one stream for the state-value function and and one stream for the action advantage function.
Policy Gradient (PG)		Decision making for network slicing resource control and management with high dimensional inputs	PG is a policy-based method and seeks to directly optimize in the policy space. There are stochastic policy gradient (SPG) and deterministic policy gradient (DPG).
Actor-Critic (AC)		Decision making for network slicing resource control and management with high dimensional inputs	In AC, a critic derives the value function (i.e. Q-value or state-value) and the actor instructs the policy distribution according to the evaluation of state-value by the critic.
Deep Deterministic Policy Gradient (DDPG)		Decision making for network slicing resource control and management with high dimensional inputs	DDPG is also called a model free off-policy actor-critic algorithm. To leverage the exploration phase, noise is added to the original policy.
Twin delayed DDPG (TD3)		Decision making for network slicing resource control and management with high dimensional inputs	TD3 is based on the actor-critic methods. It uses clipped double Q-learning with two pairs of critic networks, delayed update of target and policy networks, and target policy smoothing and noise regularization.
Proximal Policy Optimization (PPO)		Decision making for network slicing resource control and management with high dimensional inputs	PPO relies on the clipped surrogate objective function to limit having the large policy update by formalizing a constraint on the difference between the new and the old policy.

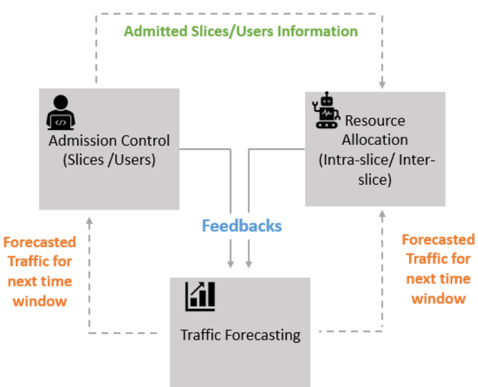


FIGURE 8. Resource management model in network slicing.

and DL-based techniques in particular, became popular in recent years to extract the spatiotemporal dependencies of mobile traffic and to leverage the automation of end-to-end resource provisioning for intelligent network slicing [64]. Moreover, one may come across some empirical studies about the superiority of DL algorithms (*e.g.*, LSTM) over ARIMA and Holt-Winters ([83], [84]). In the following, we divide the contributions on traffic forecasting in network slicing in two categories: *i)* CNN-based forecasting and *ii)* RNN and ANN-based forecasting.

To keep our discussion consistent, we use the term time interval to denote the duration covered by one sample, as part of the forecasting task, and the term time window to denote the time horizon over which the forecasting will take place. For example, if at 1 PM, we want to forecast the traffic demand for samples covering 10 minutes each, for a total of one hour, then we have a forecasting time interval of 10 minutes and a forecasting time window of one hour.

1) CNN-BASED FORECASTING

According to [85], CNN and 2DCNN have major drawbacks in recognizing temporal features, since they are specifically designed to work with images. Therefore, the authors in [86] put forward the Deepcog cost-aware network capacity forecasting framework, based on 3DCNN. In this work, historical trends of antenna-level data traffic for each base station are used as input to train the model. The cost on the operator side is associated with resource over-provisioning and SLA violations. The proposed model is applied separately for individual mobile services. The objective is to forecast the network demand based on the spatiotemporal features, so as to allow for efficient pre-allocation of resources. The authors also design a unique cost-aware loss function to train their ML framework and reduce the overall monetary cost of the operator. Ultimately, their presented framework demonstrates better outcomes than baseline methods (i.e. non-ML solutions, LSTM and autoencoder) when tested for forecasting with a five-minute time interval and a time window of up to 8 hours, in terms of resource overprovisioning, SLA violations, and overall monetary cost. It is also observed that a larger forecasting time window generally yields higher SLA violations and overprovisioning of resources. As evidence, the reported observations show a percentage of SLA violations and overprovisioning of 3% and 15%, respectively, for a given slice, when a five-minute prediction time window is used. These values reach 10% and 30%, respectively, for a prediction time window of 8h.

Similarly, the work in [87] uses 3DCNN to conduct forecasting for the same purpose (i.e. to reduce resource overprovisioning and to proactively deploy the resources to meet the future demand of slices). Accordingly, their results confirm that ML-based forecasting can achieve more than 50% reduction of monetary cost in all tested scenarios when compared to legacy non-ML forecasting methods.

The Deepcog model [86] is extended in [88] by considering multiple timescales, with both shared and dedicated resources: a long-timescale over which the instantiation of resources takes place, and a short-timescale over which resources are reconfigured. The authors propose the AZTEC framework, composed of four blocks: three blocks based on 3DCNN to forecast the resource demand for dedicated (long-timescale) and shared resources (long-timescale and short-timescale), and one block for a heuristic algorithm for resource reconfiguration. The objective is also the same as in [86]; however, their overall cost accounts for more factors (i.e. resource over-provisioning, SLA violation fees, resource instantiating fees, and resource reconfiguration fees). The numerical results suggest that the proposed framework can achieve better overall monetary cost than without forecasting for a 24-hour time window, with a long-timescale interval of 30 minutes and a short-timescale interval of 5 minutes. Besides, the authors also study the impact of dedicated and shared resource allocation strategies on the monetary cost of operators. In particular, they vary the long-timescale interval from 30 minutes to 2 hours, showing that lower intervals result in a lower cost.

2) RNN AND ANN-BASED FORECASTING

Many research works apply LSTM or one of its variants (*e.g.,* ConvLSTM) as a forecasting technique, for resource management purposes or for the automation of network slicing processes. Although it is one of the most widely used ML techniques, LSTM has a relatively high computation cost. To deal with this, some papers apply low-complexity GRU methods or simple two-layer and three-layer ANN as forecasting solutions for network slicing.

Indeed, LSTM is regarded as a promising forecasting technique for network slicing problems because of its capacity of learning spatiotemporal and long-term dependencies in data. With this motivation, [80] applies LSTM in the resource management processes to forecast the slice bandwidth demand over a time window of 200 seconds and a time interval of 20 seconds. Based on the experimental performance evaluation, the LSTM-based algorithm is shown to allow more users to access the network.

Furthermore, the authors in [89] build a collaborative learning framework combining LSTM (for large-timescale traffic forecasting with a 1 hour time interval) and A3C (for small-timescale traffic scheduling, time interval of several milliseconds) to improve resource utilization while considering a slice performance isolation constraint. The simulations show that much higher resource utilization can be achieved via the proposed collaborative framework against Q-Learning and classical AC methods, that do not consider forecasting.

In [90], LSTM, CNN and DNN are used to forecast traffic for resource management of a vehicular-specific slice, based on an SDN-enabled 5G network. The simulations show that LSTM achieves higher average forecasting accuracy (99.36%) than DNN and CNN (92.58% and 95%, respectively). Likewise, in [91], traffic forecasting using LSTM for E2E slices is conducted with a time interval of 5 seconds and a time window of 300 seconds. In this case, the accuracy of LSTM is three times higher than that of linear regression.

Again, applying the same LSTM technique but taking a different angle, the authors in [39] establish an efficient slices resource reservation strategy from the point of view of the SP (where MVNO data, i.e. aggregated traffic loads and capacity of base stations, is unknown to SP). Their proposed solution performs relatively better than the baseline ARIMA model, using as metrics the MSE, the number of over-reservations and under-reservations. Their forecasting time interval and time window are 10 minutes and 744 hours, respectively.

Aside from forecasting slice traffic, LSTM can also be utilized for other forecasting problems in the sliced cellular architecture. More specifically, to facilitate the efficient slice creation by service providers, the work in [38] uses LSTM to forecast the transmission channel condition, with a 24-hour forecasting time window and a one-hour time interval. In this work, the authors feed the output of the LSTM to a DNN, to decide whether the network can handle a new slice request or not. The proposed ML-driven approach is shown to outperform the standard analytical approach. Also, in [92], LSTM is utilized to forecast user mobility to generate the states to be dealt with by an A2C solution in charge of inter-slice resource allocation. They prove that their joint LSTM-A2C solution outperforms a GAN-DDQN solution in terms of resource efficiency in dynamic network slicing environment.

The work in [93] uses a modified version of LSTM to address the problem of accuracy in forecasting per-service traffic demands in sliced networks. The authors design an S2SConvLSTM solution, that combines the sequence-to-sequence learning paradigm and a convolutional LSTM approach. The authors achieve high accuracy for traffic forecasting over a one-hour time interval. Since S2SConvLSTM exploits the advantages of both CNN and LSTM, it outperforms them in terms of MSE and peak signal-to-noise ratio (PSNR). Additionally, the proposed method is shown to lead to good results, for five different types of services, under a variety of settings. A prediction time interval ranging from 5 minutes to 1 hour, and a prediction time window up to 3 days, were considered in the evaluations. A similar approach is proposed in [94], where the authors opt to use ConvLSTM to forecast the traffic variations on a vehicular slice with a one-hour time interval and a 100-hour time window. In a subsequent step, the authors evaluate the demand of the vehicular slice and pre-allocate the necessary resources, using primal-dual linear programming technique. They do so, under the constraint of respecting the stringent latency requirements of vehicular services.

Based on this rich literature, it is reasonable to say that LSTM and its variants have been widely used for forecasting network slice traffic and resource utilization. However, the computation cost of LSTM is relatively higher when compared to other techniques such as GRU for the same forecasting task [95]. With this in mind, the authors of [96] design a light and simplified GRU solution for forecasting resource usage per network slice over time. In their solution, the reset gate is excluded and a different activation function (softplus instead of tanh) is used in the update gate. This is shown to allow for a fast forecasting of the hourly resource usage in a network slice, while considering a time window of 120 hours. The authors propose to enforce the SLA constraints and minimize the MSE loss function in their forecasting-based network slicing resource management model. According to their experimental outcomes, the light GRU solution has much shorter computation time than LSTM due to its simpler architecture.

An even more simplified GRU approach (called soft GRU) is used in [40] for slice traffic forecasting. Soft GRU uses the same architecture as light GRU, except for the fact that input data is optimized by suppressing the historical data. As illustrated in the paper, while both soft and light GRU show the same forecasting accuracy in an hourly interval of a weekly time window, soft GRU has better computation time than its counterparts (i.e. LSTM, light GRU, and standard GRU). Equivalently, the work in [97] integrates GRU in the resource orchestrator of an SDN-based CN to predict the traffic variation of each slice in the next time interval of 1 hour. However, no evaluations are conducted for this presented framework.

In contrast to the previously discussed works, that disregard data privacy in network slicing, the authors of [98] attempt to forecast the per-slice traffic, at the base station level, while considering data privacy concerns as well as communication and computation efficiency. To this end, they rely on the Federated Proximal LSTM (FPLSTM) approach, in which slice instances (controlled by MVNOs) train local models with private datasets at the corresponding base stations, and only share trained model parameters with a global model operating on a central node (managed by InP). Hence, no data is shared among the parties and data privacy of MVNOs is guarantee. Their results show that the forecasting accuracy of their model is very close to that of a centralized model. Besides, since local models benefit from each other's knowledge, through the global model, learning rate is accelerated and notorious computation cost of LSTM is reduced.

The substantial advantages of RNN techniques, such as LSTM and GRU, cannot be disregarded when considering traffic forecasting tasks, as they are specifically designed to account for temporal dynamic behavior. However, we note that the feasibility of simple ANN solutions in the context of slice traffic forecasting was also investigated by some works. In particular, the work in [99] selects a three-layer ANN design to forecast the traffic load of each slice, with a 15-minute interval and a 24-hour time window, to proactively allocate resources in the optical transport network. The numerical results demonstrate that allocating the adequate amount of resources in advance provides better delay and lower blocking probability.

Besides, in [100], two-layer and three-layer ANNs are employed in a decentralized federated learning framework. Accordingly, service-oriented KPIs belonging to each slice are forecasted by local models (managed by SP or ST). The local models send only the extracted features to the central model (managed by InP or MVNO) for aggregation purposes. This approach protects the privacy and sensitivity of the information related to individual slices. The simulation conducted by the authors suggests that the outcomes of this FL solution are comparable to the centralized model in terms of MSE, while respecting the privacy of network slices. Besides, FL can significantly reduce the communication overhead, up to five times lower than a centralized model.

Typically, ML models for traffic forecasting are trained on large datasets, an operation which is time-consuming. To deal with this, the authors in [101] rely on TL-based DNN for traffic forecasting per slice. More precisely, they initialize the weights of their model with the weights of a pre-trained model on a similar task to perform the per slice traffic forecasting. Their TL-based forecasting model exhibits better MSE loss than a baseline ML model (where model weights are initialized randomly) while ensuring sample efficiency.

3) LESSONS LEARNED

It is quite obvious from the literature that forecasting mobile traffic demand brings significant benefits to resource management and QoS-oriented mechanisms, enabling an increased automation in the network slicing process [96]. Therefore, numerous DL-based forecasting solutions have been proposed, as detailed above. It is noteworthy that

TABLE 3. ML-based forecasting techniques in network slicing. "NA" means the required information is "Not Mentioned" in the paper itself.

Ref	Focus	Forecasting technique	Forecasting item	Time interval	Time window	Evaluation method
[86]	Minimize resource overprovision and SLA violations	3DCNN	Total traffic per slice	5 minutes	8 hours	Real-world dataset from video streaming (11 eNodeBs) , Snapchat (70 eNodeB), Facebook (470 eNodeB).
[88]	Minimize overall monetary cost of operators	3DCNN	Total traffic per slice	5-30 minutes	24 hours	Real-world dataset from video streaming (11 eNodeBs), Snapchat (70 eNodeB), Facebook (470 eNodeB)
[87]	Miminize resource overprovision and SLA violations	3DCNN	Total traffic per slice	NA	NA	Real-world dataset from video streaming (11 eNodeBs), Snapchat (70 eNodeB), Facebook (470 eNodeB)
[80]	Maximize user acceptance	LSTM	Bandwidth requirement per slice	20 seconds	200 seconds	Traffic dataset is generated using iperf3
[89]	Maximize resource utilization	LSTM	Total traffic per slice	one hour	NA	Traffic dataset is generated based on Gaussian distribution
[90]	Maximize forecasting accuracy	LSTM, CNN, DNN	Total traffic per slice	NA	NA	Traffic dataset is generated using light-weight fork of mininet emulator
[91]	Maximize throughput, delay and link utilization	LSTM	Total traffic per slice	5 seconds	300 seconds	Traffic dataset is generated using iperf
[39]	Minimize resource reservation made by Service Provider	LSTM, DNN	Total traffic per slice	10 minutes	72 hours	Real-world data is collected from 15 base stations owned by a major MNO within the city of Shanghai, where data is recorded for 1 month period
[38]	Maximize SLA requirements	LSTM	Transmission channel and per slice admission decision	one hour	24 hours	Traffic dataset is generated using ATERR mechanism in the NS-3 Network Simulator, simulated scenario recorded 24 hours
[92]	Maximize resource utilization	LSTM	Track the user mobility to create the network state	NA	NA	Traffic dataset is generated from a simulation over an area of 240m × 240m, with 1200 UEs within the same slice sharing the moving pattern
[93]	Mimimize MAE	S2SConvLSTM	Total traffic per slice	5, 30,60 minutes	3 days	Real-world traffic dataset is generated from 792 antennas aggregated every 5 minutes
[94]	Minimize latency	ConvoLSTM	Total traffic per slice	one hour	100 hours	Real-world traffic dataset is collected from Milan, Italy for three types of services SMS, phone and web traffic
[96]	Minimize loss function and maximize SLA	Light and simplified GRU	Resource usage per slice	one hour	120 hours	Real-world traffic dataset is from a live cellular network recorded over five days for sites located in a dense urban area
[40]	Trade-off between resource over-provisioning and slices isolation	Soft GRU	Total traffic per slice	one hour	3 days	Real-world traffic dataset is from a live cellular network recorded over five days for sites located in a dense urban area
[97]	Minimize reconfiguration penalties and the power consumption	GRU	Total traffic per slice	one hour	NA	Real-world traffic dataset is collected from MBB service of Milan, Italy
[98]	Minimize RMSE	FPLSTM	Total traffic per slice per base station	10 minutes	2 days	Real-world traffic dataset is collected from 57 eNodeB of Orange network, France
[99]	Maximize prediction accuracy	three-layer ANN	Total traffic per link	15 minutes	24 hours	Real-world traffic dataset is from optical network of China Telecom Corporation
[100]	Minimize services degradation	three-layer ANN, two-layer ANN, FL	Service KPIs per slice	NA	NA	Traffic dataset is generated using simulator based on Open-AirInterface (OAI) platform
[101]	Minimize MSE	TL-based DNN	Total traffic per slice	one hour	one week	Real-world traffic dataset is collected from commercial 5G (Sub-6 GHz and mmWave) networks

different forecasting time intervals, associated with different time windows, are considered in the existing state of the art.

Generally, we observe that the duration of the considered time interval ranges from a minimum of several seconds to a maximum of one hour. On the other hand, in terms of the time window, most of the works consider a minimum of several seconds and a maximum value in the order of weeks. This is a consequence of the number of problems that can benefit from the forecasting function, which are diverse in terms of requirements. For example, large reconfiguration periods (i.e. minutes or hours time interval) are acceptable for VNF dimensioning in the CN (see for instance [40], [86], [88], [96], [97]), while forecasting of traffic in the RAN needs to cover the requirements of the radio resource reconfiguration process (i.e. below-second time interval, see for instance [80], [89], [91]). Also, while mobile traffic demand shows significant periodicity at a daily and weekly time scales, it presents significant dynamics at the minute time scale. This means that using a large forecasting time window would be adequate in some cases, and using a smaller one would be adequate in others.

All in all, it is sensible to stress that, in general, integrating the forecasting function in the network slicing framework leads to better outcomes. Notably, LSTM is a quite popular method, and its modified version S2SConvLSTM outperforms CNN, 3DCNN, LSTM in terms of MAE and PSNR values. It is also worth noting that GRU shall be a better choice if one is looking for a lower computation time than LSTM. Overall, each of these forecasting methods has its benefits and its disadvantages. To the best of our knowledge, none of the papers explicitly answers when to use one forecasting technique over the others. Therefore, it is hard to say that one unique method is universal. Table 3 gives a comprehensive summary of all contributions on ML-driven forecasting in network slicing.

B. ADMISSION CONTROL

Admission control systems operate at two levels: slice admission and end-user admission. Slice admission control is relevant to the InP, whereas end-user admission control is relevant to the MVNO [13]. Recently, some research works also study two-level admission control systems whereby both slices and end-users admissions are covered simultaneously [102]. We present accordingly contributions on admission control, under the following three categories: (i) Slice Admission Control (ii) End-User Admission Control and (iii) Two-level Admission Control.

1) SLICE ADMISSION CONTROL

Slice admission control can be event-driven or periodic [103]. More specifically, as slice requests arrive in the system, operators might want to trigger a slice admission decision (i.e., reject or accept) immediately upon slice arrival [80]. Alternatively, operators might want to manage decisions periodically by holding the slice requests in a queues [87]. Both the event-driven and periodic slice admission problems can be modeled as MDP [103].

In terms of event-driven admission control, the authors in [27] model the RAN slice admission control problem as sMDP, to deal with the stochastic arrival of slices on the go. They consider two types of slices: inelastic slices, that require constant throughput, and elastic slices, that do not require a constant throughput. The authors utilize the Q-learning method for deriving decisions. They do so with the objective of maximizing the long-term revenue, by choosing an adequate action (i.e. accepting or rejecting slices). As shown in the simulation results, the algorithm tends to admit more inelastic slices (which generate higher revenues) than elastic slices and yet provides a better reward level than other benchmark solutions. Similarly, the work in [104] models the slice admission problem in a fog-enabled network using sMDP. The authors use DDQN to solve the problem. Their results show that between 10% to 60% higher revenues can be obtained, with respect to baseline methods.

In [105], the authors also consider an event-driven slice admission scheme, under a limited network transmission capacity, in the context of Next-Generation RAN (NG-RAN). More specifically, their problem is based on the model-aware MAB problem, with two additional constraints, taking into account the system capacity and the slice life cycle. To solve the problem, the authors rely on a low-complexity enhanced UCB algorithm to select the slices for admission, according to resource multiplexing gains. To validate the effectiveness of the proposed framework, the authors run the proposed algorithm in an LTE commercial system. Their model is shown to allow accepting more slices and enabling a better system utilization, when compared to random and greedy approaches. When it comes to periodic admission control schemes, slices requests are buffered in a queue over a given period of time. Requests are then processed sequentially, at the end of the period [103]. Accordingly, the authors in [79] employ a DQN approach which observes the system queue length and resource availability and aims at maximizing the resource utilization and minimizing the queue length. Their simulated results reveal that the DQN is superior to Q-Learning, Greedy, and Random algorithms, in terms of average utility for each service request.

Yet, in practice, simply admitting slices in a periodic admission control scheme would not help to increase the profits of the operator [81]. In fact, during the periodic servicing time, slices requirements may vary and SLA violations could occur if the operator doesn't satisfy potential additional slices resources requirements on time. With this consideration, the work in [106] exploits RL for slice admission control processes, in a C-RAN architecture. The objective is to reduce the penalty fees, incurred by the violation of SLAs. Specifically, the RL agent is designed to learn the relationship between slice acceptance/rejection and the incurred overall profits. The findings exhibit that RL provides a significant reduction in losses of the operator (64 to 80 % reduction for low network loads and 14 to 39 % reduction for high network loads), with respect to a benchmark model, that admits slices whenever resources are available. With the same spirit, in [107], the authors use the policy-based RL to derive periodic slices admission decisions. They do so, while considering obtained reward parameters (i.e., a sum of all rewards from already admitted slice and potential new slice admission). The evaluation shows that RL achieves approximately 75% to 30% reduction in penalty for a slice, than baseline approaches.

Equivalently, the authors in [108] establish a network slice admission and congestion control system, integrated with the 3GPP network slice deployment framework [109]. Their ultimate goal is to maximize the resource utilization, while reducing the blocking probability of high priority slices. Eventually, this allows to maximize as well the InP revenue. To do so, the authors map the incoming slice requests from the queue into NSIs, which are composed of a set of NFs spanning across the RAN, TN, and cloud. For slice admission, SARSA is integrated with Linear Function Approximation (LFA) to solve the MDP. The proposed approach is shown to present superior results to the greedy approach, in terms of long-term reward and slice blocking probability.

While some papers are focusing only on the admission control part, other papers are coupling the admission control function with the forecasting or resource allocation function. We note that periodic decision models are those combined with the forecasting and resource allocation functions. Accordingly, the empirical study in [77] suggests that forecasting-aware admission control outperforms admission control function without forecasting. However, this comes at the cost of slightly longer computation time (approximately 3514 seconds for 30 slice requests, which remains acceptable in overall system implementation). In [80], the authors consider a heuristic-based admission control approach, coupled with an LSTM-based traffic forecasting method. A higher user acceptance rate is obtained with the proposed scheme, resulting in approximately 18% more revenues to the InP, with respect to a baseline scheme.

The authors in [110], jointly consider the problems of slice admission and resource allocation, with a focus on beyond 5G RAN, with a cell-free mMIMO setup. They assume a single type of VNF exists per slice and rely on a modified deterministic actor-critic algorithm (called D-TD3 with state-action distribution function) for solving the problems. Their objective is to do so, while minimizing the network deployment cost. The reward clipping mechanism is utilized in this work to avoid destabilization in the training period. The simulation results suggest that, in general, D-TD3 carters better results than baseline methods (i.e. DDPG, Stochastic AC, TD-3) in terms of average return level, admission rate, CPU utilization, average delay, and average power consumption. With the same objective as in [110], the works in [111], [112], and [113] formalize the joint slice admission and resource allocation problem, for network slicing, in a 5G C-RAN network. These works consider more than one VNF per slice. They apply the TD-3 method [111], an enhanced TD-3 approach [112] and a multi-agent PPO technique [17] (i.e. Slicing agent[1] and AC agent)[2] to achieve a more stable and faster learning process and to meet the desired objectives. The proposed approach is shown to outperform other baseline methods (i.e. DDPG, Stochastic AC and Greedy approaches). Similarly, the authors of [114] and [115] rely on the DQN and DDQN to establish a joint slice admission and resource allocation framework to achieve higher InP revenues and QoS. Their presented mechanism outperforms Q-Learning and Greedy approaches in terms of long-term revenues and user acceptance rates.

2) END-USER ADMISSION CONTROL

As for the end-user admission control problem, it is commonly assumed that slices are already deployed in the network, with the slices information (i.e. capacity of slices) given. Both supervised learning approaches and RL approaches are used for solving end-user admission control problem.

The work in [116] attempts to map traffic flows of end-users to pre-deployed slices, by utilizing a supervised learning method. Precisely, the authors resort to the edge-based GCN method to predict to which slice a traffic flow request should be admitted. They aim to do so, while maximizing successful transmission of requested traffic, over network slices. Their results show that GCN-based approach outperforms random, round-robin, and Multi-Layer Perceptron (MLP) methods in terms of successful transmission rate and amount of transmitted data.

In [117], the authors exploit supervised learning and unsupervised learning techniques, namely random forest and DNN techniques, to map incoming user requests to the appropriate network slices. They derive their decisions, with the objective of enabling load balancing among slices. Random forest is applied to classify well-structured data (i.e. network KPIs), and DNN is used to classify unstructured ones. Besides, a master slice is considered in this work. In case a non-master slice (i.e. eMBB or URLLC or mMTC) fails or gets overloaded, end-users traffic could be redirected to the master slice. Based on the results, the mechanism is shown to address slice failure issues, while predicting with a high accuracy the slice types onto which users should be admitted.

While previously discussed works apply supervised and unsupervised approaches for end-user admission control, the study in [118] applies reinforcement learning, and in particular DQN, to admit users in real-time. The objective is to maximize MVNOs profits. Accordingly, the problem is modeled as sMDP. The results show that DQN outperforms Q-Leaning and random approaches, in terms of system utility and average system throughput.

The works in [119] and [120] study the problem of end-user admission, while considering two types of users (i.e., eMBB users and vehicular users). Users are to be admitted to a hotspot slice instance and a vehicular-to-infrastructure (V2I) slice instance, in a fog-RAN setting. To solve the problem, the authors rely on a DQN agent [119] and a Q-Learning agent [120]. Their objective is to maximize the reward, encompassing content caching performance in the fog-RAN, as well as throughput and delay over the slices. Based on the simulation results, the proposed mechanisms are shown to be superior to other baseline methods, in terms of cumulative reward, associated with their objective functions.

Similarly, the authors in [121] address the end-user admission problem, and rely on a DRL approach to solve it. In particular, the ensemble learning method (ELM), which exploits the benefits of SPG and Approximation Framework (AF), is used. The authors compare the performance of SPG and AF and show that, as expected, SPG achieves higher rewards than AF, while AF converges faster than SPG. ELM is also shown to outperform SPG and AF in terms of resource block (RB) utilization and user admission rate.

3) TWO-LEVEL ADMISSION CONTROL

All works in the previous two subsections tackle one of two problems of slice admission or end-user admission. None of them considers the two problems jointly. In fact, by considering the requirements of both slices and users, it is possible to build efficient two-level admission control systems [102].

The authors in [122] formulate the two-level admission control problem by considering both the slice and end-users to satisfy the end-users QoS requirements while ensuring the isolation among slices. In this regard, they utilize a heuristic-based Jacobian (J matrix) to monitor the violations of predefined KPIs and resolve them by adjusting the corresponding control parameters (i.e. weight of the slice, capacity limit of the slice) iteratively. Consequently, DQN is deployed to find the near-optimal value from the scratch based on the control parameters and KPIs status of each service. On top of that, they establish the hybrid of J matrix and DQN where the J matrix changes the control parameter based on the decisions made by the DQN agent. According to their assessment, the hybrid model provides the nearest optimal value (approx 0.1% gap with optimal value) over other baseline methods.

The same problem is tackled in [123], with another approach. The authors consider a three-layer ANN to estimate the J matrix, based on the control parameters and network conditions. The results show that the results of the J matrix and ANN are comparable.

4) LESSONS LEARNED

Indeed, admission control is one of the main building blocks of network slicing. Incorporating ML techniques to this

[1]Slicing agent ensures the efficient slice embedding to the substrate network.

[2]AC agent ensures the slice instances are admitted by means of maximising the InP revenue.

TABLE 4. Summary table of ML-based admission control in network slicing. Column titles: "Input," "Decision" and "Objective" are relevant only for the papers which use combination of DL and heuristics. "NA" means the required information is "Not applicable."

Scope	Ref	Focus	ML Technique	Model	State/Input	Action/Decision	Reward/Objective
Slice Level	[27]	Maximize long-term revenue of InP	Q-Learning	sMDP	Number of slices and new slice arrival and departure	Admit or reject slices	Prices per unit time paid by slices
	[104]	Maximize long-term revenue of InP	Dueling DQN	sMDP	Slices request arrival, available InP resources and Fog resources	Admit slice with borrowing resources from tenants, Admit slice w/o borrowing resources from tenants or reject slice	Reward associated to taking action
	[105]	Maximize multiplexing gain	enhanced UCB algorithm	model-aware MAB	Arms are considered as slices	Admit or reject slices	Resources asked by slices and multiplexing gain
	[79]	Maximize resource utilization and minimizing queue length	DQN	MDP	Queue length and available resources	Admit or reject slices	Successfully serving requests minus the cost of the queue delay
	[106]	Maximize overall profits	ANN	MDP	Resource utilization and slice requirements	Admit or reject slices	Overall profits minus penalty cost
	[107]	Maximize overall profits	Policy-based RL	MDP	Slice requests	Admit or reject slices	Total profits from already admitted slice and potential new slice admission
	[108]	Maximize overall profits	SARSA	MDP	Queue length and available resources	Admit or reject slices	Profit for accepting slices and loss for dropping slices
	[80]	Maximize acceptance of slice and resource utilization	LSTM-Heuristic	NA	UE new traffic flow requests	Admit or reject slice	To maximize the number of acceptance slice and resource utilization
	[38]	Maximize SLA requirements of slices	LSTM-DNN	NA	Characteristic of wireless transmission channel	Fulfill new slice SLAs or not	To enhance SLA requirements of slices
	[110]	Minimizing latency, energy consumption and VNF instantiation cost	D-TD3	MDP	Number of slice arrivals, allocated CPU resources, the overall delay, energy status and the number of users per slice	Scaling up and down of CPU and beam-forming power	SINR threshold and penalty cost for violating the pre-defined constraints
	[111]	Minimizing latency, energy consumption and VNF instantiation cost	TD-3	MDP	Number of new UEs, computing resource allocation to each VNF, number of UEs being served, delay and energy status, VNF instantiation cost	Scaling up and down of resources	Total network cost
	[112]	Minimizing latency, energy consumption and VNF instantiation cost	enhanced TD-3	MDP	Number of new UEs, computing resource allocation to each VNF, number of UEs being served, delay and energy status, VNF instantiation cost	Scaling up and down of resources	Total network cost
	[113]	Maximize InP revenue	Slicing agent-PPO AC agent-PPO	MDP	Type of slice request, operational time offered revenue, substrate network status	Slicing agent: slice embedded or not AC agent: admit or reject slice	Slicing agent: Positive reward if slice is embedded and accepted AC agent: Positive reward if slice is accepted
	[114]	Maximize the revenue of InP and quality of end-to-end services	GRM: DQN LRM: DQN	MDP	GRM: The number of accepted slice requests LRM: QoS of admitted slices	GRM: Admit or Reject Slices LRM: Permit slice adaptation or not	GRM: Average slice utility LRM: Net profits of InP
	[115]	Maximize the number of accepted slice	DDQN	sMDP	Number of slices in a given class and new slice request	Admit or reject Slice	Total Slice acceptance
User-Level	[116]	Maximize successfully transmission of requested traffic flow	GCN	NA	Traffic flow requests	Flow requests mapping to appropriate slice	To improve transmitted success rate of traffic flows
	[117]	Maximize network utilization and availability	Random Forest-DNN	NA	Users' devices KPI datasets	Slice selection	To enhance network utilization and availability
	[118]	Maximize overall profits	DQN	sMDP	Resource weight assigned to the users by slice	Admit or reject users	Payment from Slice Tenant
	[119]	Maximize caching performance and throughput and minimize delay	DQN	MDP	Channel coefficient, residual bits and slots, cache feature, cache state, current request	Content selection to cache and mode selection of users	Constraints satisfaction and cache hit rate
	[120]	Maximize network performance	Q-Learning	MDP	Current mode selection of UEs	User selection of fog-RAN or RRU	Constraints satisfaction and power-minus-rate function
	[121]	Maximize the RB utilization and minimize the operation cost	Ensemble Learning Method	MDP	User throughput fulfilment rate, request error rate of users, RB availability ratio	Increase or decrease weight of resource allocation, modification, threshold of scale-in and scale-out	Reward associated to taking action
Two-level	[122]	Maximize resource fulfillment satisfaction and minimize interference among slices	DQN	MDP	Network control parameters and KPI report	Increase or decrease or keep the control parameter	KPI satifaction
	[123]	Maximize multiplexing gain and minimize interference among slices	three-layer ANN	NA	Network control parameters and network condition vector	Network KPIs	To improve multiplexing gain and interference among slices

end further helps operators maximize their revenues, while ensuring SLAs. In fact, admitting a maximum of users/slices can help maximize the long-term operator revenue [124]. Nevertheless, by admitting more users/slices, Key Performance Indicators (KPIs) may degrade, leading to SLA violations and potential penalty fees [81]. Thus, it is essential to maintain the balance between resource utilization and KPIs values [27]. It is worthwhile to stress that these aspects could be covered by a proper reward function design in an MDP/MAB model.

Furthermore, it is worth noting that supervised learning was commonly used for end-user admission control, while model-free RL methods are commonly applied for slice admission control and two-level admission control. Overall, MDP formulation was mostly used throughout these works. More specifically, states usually represent the number of accepted slices/users, actions represent the acceptance or rejection of slices/users', and rewards are mostly evaluated as the overall profits of operators. Finally, as of our knowledge, ML-based techniques used so far for slice admission control problems are still in their infancy and need further explorations and verification. For ease of reference, all the existing works related to ML-driven admission control in network slicing are summarized in Table 4.

C. SLICE RESOURCE ALLOCATION

In recent years, a large number of studies have applied ML techniques to solve complex resource allocation problems in network slicing. These problems target the different network domains, where different types of resources are implied. For instance, spectrum (a.k.a resource blocks (RBs), frequency-time blocks) and transmit power are the main resources in the RAN domain. Wavelength and bandwidth are the main resources in TN. Finally, VNFs, CPUs, and memory are the major one in CN domain.

In this section, we review contributions on network slicing resource allocation with ML techniques, while considering the following categories: (i) resource allocation in RAN, (ii) resource allocation in CN, (iii) resource allocation in TN, (iv) resource allocation in RAN and CN and (v) end-to-end resource allocation.

1) RESOURCE ALLOCATION IN RAN

A considerable amount of works on network slicing targets resource allocation in RAN, by exploring various ML-based techniques. RAN domain slicing implies RAN resources (i.e. spectrum, transmit powers, etc.) sharing and allocation [125].

The resource allocation scheme can be coarse-grained or fine-grained. In a coarse-grained resource allocation approach, resources are provisioned to slices by considering merely average slice-level QoS requirements (i.e. average slice throughput, average slice latency, etc.) and without taking into account any end-user level requirements. Instead, a fine-grained resource allocation approach allocates resources (i.e., Resource Blocks (RBs), CPU and transmit power, etc.) to the UEs of each slice by taking into account the end-user QoS satisfaction level. Technically, the LTE subframe is considered for RBs allocation [126]. It includes 12 consecutive sub-carriers (i.e. 180kHz) per RB in 1-ms Transmission Time Interval (TTI). It is commonly assumed in papers that RBs are shared through the Orthogonal Frequency-Division Multiple-Access (OFDMA) method for downlink (DL) transmission to address the interference issues among UEs [127].

Notably, traffic variations at slice-level can be observed over time intervals in the order of hours/days. Instead, users traffic variations can be observed over time intervals in the order of minutes/seconds [128]. Accordingly, large timescale is generally considered for coarse-grained slice-level resource allocation and small timescale is considered for fine-grained user-level resource allocation.

While most of the RAN resource allocation contributions introduce a coarse-grained or a fine-grained approach, a few introduce two-level resource allocation schemes with both coarse-grained and fine-grained approaches. Thus, we group contributions under the following categories: (i) Coarse-grained Resource Allocation (ii) Fine-grained Resource Allocation, and (iii) Two-level resource allocation.

Coarse-grained Resource Allocation: Both RL and supervised/unsupervised methods are employed for coarse-grained resource allocation. As mentioned earlier, coarse-grained resource allocation targets average slice-level requirements. In this regard, the work in [129] relies on the DQN-based algorithm to tackle the inter-slices bandwidth allocation problem, while considering high traffic variations among slices. A discrete normalized advantage function (DNAF) is integrated into DQN to leverage faster convergence to cope with a larger action space. The numerical results show the superior performance of DNAF-based DQN over classical DQN. However, the proposed DNAF-based approach could not balance between QoS and spectrum efficiency (SE) and that calls for further investigations. Attempting to investigate the trade-off between QoS and SE, the authors of [130] deploy DQN to allocate bandwidth resources to slices, while considering 25 use cases of NGMN [6]. Interestingly, the user satisfaction score is above 80% for all the use cases with minimum bandwidth allocation. However, no comparisons are done with existing counterparts.

The authors in [131] study the same problem as in [129]. They employ a Generative Adversarial Network (GAN)-based DDQN approach [132] to have a better estimation of expectation of state-action values (i.e. Q-value). By that, they aim to overcome potential oscillations in the Q-value estimation [133], that exist with classical RL techniques. Besides, for more stability in the training process, they then utilize the reward clipping mechanism (a.k.a reward reshaping), attributing a value to the reward in $[-1, 0, 1]$, according to specific constraints. Through numerical results, it can be seen that the proposed framework provides better results in terms of system utilization, with respect to classical DQN. Moreover, with reward clipping, their system is shown to converge with a higher system utilization.

Exploiting the benefits of dual connectivity (DC),[3] the work in [134] formulates the multi-slice resource allocation problem in RAN, which is composed of macro and small cells. To this end, they introduce the dueling double deep Q-network with LSTM (LSTM-D3QN), in which network state is inferred and yet leverage efficient mapping from state to action. Their numerical results show that their solution is superior to its counterparts (i.e. LSTM-A2C and DQN) in terms of resource utilization and QoE. Besides, DC-based network slicing shows better resource efficiency and QoE than the one without DC.

Coarse-grained resource allocation has also been investigated for network slicing in the RAN domain, for specific use cases (i.e. smart grid and V2X use cases).

The work in [135] focuses on the bandwidth allocation to RAN slices to serve various types of smart grid traffic, with the objective of maximizing the utility of bandwidth resources. They assume slices serve two types of applications: elastic applications (not sensitive to bandwidth requirements) and real-time applications (with a minimum requirement on QoS to satisfy). Then, they rely on a DQN approach to allocate bandwidth to slices, while maximizing a reward function, combining spectrum efficiency and the utility of slices. Moreover, they consider a double DQN approach and compare the convergence rate of both approaches in terms of reward value. Both DQN and double DQN show comparable results.

Similarly, the authors in [136] study coarse-grained slices resource allocation for V2X services. They use simple Q-Learning to maximize the resource utilization of multiple slices in the network. Their Q-Learning approach is shown to be superior to a fair and a greedy resource allocation approaches, in terms of utility score. Likewise, the work in [137] targets resource allocation for multiple slices, offering services to vehicular and smart cities' users. DQN is used to derive decisions, on whether request tasks will be served with edge or core resources. The numerical results

[3]Dual connectivity allows UE to access the resources from different eNodeBs simultaneously.

TABLE 5. Summary table of ML-based resource allocation in network slicing. Column titles: "Input," "Decision" and "Objective" are relevant only for the papers which use a combination of DL and heuristics. NA" means the required information is "Not applicable."

Scope	Ref	Focus	ML Technique	Model	State/Input	Action/Decision	Reward/Objective
RAN (Coarse-grained)	[129]	Maximize the long-term spectrum efficiency and QoE	DNAF-based DQN	MDP	Traffic variation per slice	Bandwidth allocation to slice	Combination of spectrum efficiency and QoE
	[130]	Maximize resource utilization and QoS	DQN	MDP	SINR, latency, arrival rate, error rates and packet size	Bandwidth allocation to slice	User satisfaction ratio
	[131]	Maximize the long-term spectrum efficiency and QoE	GAN based DDQN	MDP	Traffic variation per slice	Amount of bandwidth allocated per slice	Combination of spectrum efficiency and QoE
	[134]	Maximize the long-term utility and QoE	LSTM-D3QN	MDP	Number of successful transmitted packet	Amount of bandwidth allocated per slice	User satisfaction for throughput and QoE
	[135]	Maximize the long-term utility of bandwidth resources	DQN	MDP	Traffic Volume per slice	Amount of bandwidth allocated per slice	Combination of spectrum efficiency and utility models of applications
	[136]	Maximize the resource utilization efficiency	Q-Learning	MDP	Number of slices requests	Resource allocation to slice	Overall utility of slices
	[137]	Maximize the grade of service (GoS) and utilization	DQN	MDP	Slice status, available resources of fog node and current load of fog node	Decision of whether Fog node or cloud serves the slice	Reward for accepting or rejecting of high-utility and low-utility slice
	[94]	Minimize system delay	ConvoLSTM-heuristic	NA	Historical service Data	Resource distribution of each slice	System delay
	[138]	Minimize the long-term network operation revenue	DQN-CNN-LSTM	MDP	Slice resource status, QoS level and vehicular traffic density and number of VUEs per slice	Adjustment of network slicing configuration scheme	Combination of latency, reliability requirements and cost function of network resources
	[139]	Maximize system throughput	Supervised MLs	NA	Slices' traffic profile and slice blueprints	Slices' requirements classification	To enhance system throughput
RAN (Fine-grained)	[140]	Minimize the latency	Enhanced MAB algorithm	MDP/MAB	Slicing PRB resource configuration for different slices	Slices resource allocation ratio	Latency status of each slice
	[141]	Maximize the network utility	Q-Learning	MDP	Resource availability and slice request	Resource allocation to slice	User resource requirement satisfaction
	[142]	Maximize the long-term RBs utilization	Offline Q-Learning	MDP	Slice status and number of UEs in the network	Slice resource allocation ratio	Normalized resource utilization
	[143]	Minimize the violations of URLLC constraints and rate violations threshold of eMBB users	Four-layered DNN	NA	Reported CSI	Beamforming weights	To improve delay and rate violation threshold of URLLC and eMBB users
	[144]	Minimize the overall energy cost of system	DNN	NA	Wireless channel gains, tasks' deadline	Set of selection slice	To improve the overall energy cost of system
	[145]	Minimize inter-numerology interference	DQN	MDP	Sub channel gain of each user and INI power status on each sub channel	Feasible subchannel allocation to slice	Overall throughput of users associated to slice
	[146]	Maximize the slice utility	DQN and A2C	MDP	Graph Attention Network (GAT): Spatial and temporal correlation and variations of base stations	Bandwidth allocation to slice	Combination of QoS satisfaction and resource efficiency
	[147]	Maximize the slice satisfaction and minimize RB allocation	DQN based Ape-X	MDP	Network slice satisfaction ration, RB utilization ration and slice status	Increase or decrease or no change of number of RBs allocated to slice	Combination of network slice satisfaction and resource block utilization
	[148]	Minimize operational cost and maximize QoS	SAE, AC and DNN	CB	Traffic arrivals, Mean and variance of SNR	RBs, MCS and CPU scheduling policies	Combination of performance, decoding error probability and resource usage
	[149]	Minimize OPEX of InP	DNN	NA	Services Traffic, CQI, MIMO full-rank, PRB, CPU consumption, RRC connected users	Number of DL PRBs allocated to slices	OPEX of InP
	[127]	Maximize the end-users throughput	B&C and LSTM	NA	RAN conditions, Fronthaul and computing capacity	Optimal RAN resource allocation at user level	End-users throughput
	[150]	Maximize the QoS: energy and queue delay	SAC	cMDP	Channel, battery and queue length status	Channel selection and energy harvesting time	Total throughput of users for all the slices in the system
	[151]	Maximize slice throughput	Adaptive Interior-point Policy Optimization (IPO)	cMDP	Number of users in each slice	Bandwidth assignment for each type of users	Total throughput of all slices
	[152]	Maximize the profit of service provider	DQN	MDP	Location of user, task arrival of user and queue size of user	Decision of computing offloading and packet scheduling	Combination of utility of users per slice and slice payment
	[153]	Maximize the overall utility of slice	DQN	MDP	Number of slices requests and energy efficiency of slices	A set of feasible action (i.e. Transmission Power and Spreading Factor selection for slice)	Throughput and delay of devices associated to slice

show that, with respect to benchmark solutions, the proposed approach presents the highest performance in terms of reward (i.e. encompassing utility, grade-of-service (GoS) and cloud avoidance).[4]

As V2X services are characterized by a high mobility and spatiotemporal correlations [154], the authors in [94] rely on the integration of ConvoLSTM with primal-dual interior-point method to forecast the complex slice traffic and allocate resources to slices accordingly. The proposed framework is shown to offer better resource utilization with respect to a resource allocation framework that does not include forecasting. In addition, they show that the proposed framework can better meet overall system delay requirements, since resources can be allocated in advance. Similarly, the study in [138] also focuses on V2X traffic and considers correlations over space and time for allocating resources. More precisely, the authors combine DQN and CNN to capture spatial dependencies and LSTM to extract temporal dependencies, to enable an efficient resource allocation. Their results show that their network slicing framework provides lower blocking error rate (BLER) and latency than the baseline approach.

[4]Cloud avoidance values represent the level of edge nodes contribution, i.e. a higher cloud avoidance ratio, reflects a better efficiency of edge nodes.

While previously discussed works on coarse-grained slices resource allocation employ RL to this purpose, the authors in [139] rely also on supervised learning approaches to derive decisions. In particular, they introduce a network slice resource management orchestrator, encompassing a ML-based classifier, a ML-based predictor, an admission control function, a slice scheduler, and a resources manager. Firstly, they employ various ML-based Classifiers (i.e. KNN and SVM) to classify network demands (including SLA) requested by a service provider. Then, a Regression Tree (RT) method is used to forecast each slice resources ratio, serving as input to the admission control module. From then on, they rely on heuristic approaches for both admission control and slice scheduler. Based on the evaluations, RT (i.e. complex, medium, and simple) is shown to provide a better MSE compared to its counterparts, in some cases even six times lower. Besides, their framework leads to a very small gap (approximately 5%) to optimal values, in terms of slice ratios, while static and random approaches imply gaps of 25% and 35%.

Fine-grained Resource Allocation: Similarly to coarse-grained resource allocation, RL, supervised and unsupervised methods have been applied for fine-grained resource allocation. As mentioned earlier, the fine-grained resource allocation approach considers the end-user QoS satisfaction level. The majority of works consider the RBs allocation to end-users, while some attempt to allocate other types of resources (i.e. CPU and energy/transmit power).

The authors of [140] allocate RBs to end-users in eMBB/URLLC slices to meet stringent latency requirements. Towards this end, their problem is first characterized as MDP and then transformed into a model-aware MAB problem. They defined the bandit's reward function based on the current system configuration, channel quality, and slice traffic demand. Moreover, they opt for an enhanced MAB algorithm that exploits the advantages of both UCB and TS. Their simulations illustrate the superiority of their proposed algorithm over UCB and TS in terms of latency, buffer size and SNR. Likewise, the work in [141] employs the Q-Learning method in the dynamic RB allocation to users associated with eMBB, URLLC, and mMTC slices. Their results conclude that approximately 35.6% better resource utilization can be obtained, compared to a random scheme.

Equivalently, the authors in [142] attempt to investigate dynamic RB allocation to different types of slices (i.e. eMBB and Vehicle to Vehicle (V2V)) in NG-RAN. Resource allocation for eMBB is considered for both uplink (UL) and downlink (DL) directions. However, V2X communication can be done through either base station (using UL/DL) or via nearby vehicles (using SideLink (SL)), as introduced in 3GPP release 14 [155]. While considering these aspects in the system model, the authors aim to maximize the RBs utilization, by using a Q-Learning agent. Their results show that their proposed approach outperforms the fixed resource slicing scheme in terms of RB utilization, latency, data rate, and service outage. Moreover, paying particular attention to the vehicular user equipment (VUE), the works in [143] and [144] study RB allocation with respect to multiple slices, by accounting for the channel gain of the vehicular network environment. In this regard, they utilize DNNs to extract features from non-linear relationships among VUEs, thereby finding the optimal resource assignment policies. It is learned from their numerical results that having a good knowledge of channel gain can reduce the radio resources overhead cost to 50% [143] and yet energy efficiency is approximately 26% better than baselines in a certain scenario [144].

Notably, only a few studies shed light on interference when studying coarse-grained RB allocation. The authors of [145] study the RB allocation to end-users in individual slices, with the objective of minimizing the inter-numerology interference (INI)[5] among slices. To solve the problem, the authors rely on a DQN agent, trained offline, and later invoked online to allocate RBs, while considering dynamic users requirements. Their evaluations show that their DQN-driven framework allows to minimize INI, with results close to the optimal solution.

[5]Numerology of spectrum channels in 5G is important to enable flexibility in offering diverse services, yet in turn it introduces a new type of interference (named INI) among slices [156].

As cooperation among base stations can help mitigate interference [157], the authors in [146] develop an online resource management framework that relies on DQN and A2C agents. In their framework, the authors utilize GAT to extract the spatiotemporal correlation among base stations. This correlation is further used as a representation of network states, and fed to the DQN and A2C agents, to find the optimal spectrum bandwidth allocation policies. Through their simulations, it is clearly seen that GAT-driven DQN and A2C can improve the spectrum efficiency, while reducing the interference among base stations.

While majority of papers (see for instance [140], [141] [142], [145] [146] just to name a few) investigate dynamic RB allocation with a fixed number of slices, the authors in [147] study RB allocation to UEs, under a varying number of slices. Their adaptive algorithm is inspired by the distributed learning Ape-X RL technique [158] where multiple actors are used for multiple instances of the environment. Basically, Ape-X is a modified DQN method, with multiple actors enabling parallel resource allocation to multiple slices simultaneously. Overall, each agent focuses on the efficient RBs allocation to satisfy each UE requirement of the slice. Their results prove that their proposed scheme can allocate the RBs steadily, even if the number of slices varies.

As previously mentioned, some papers focus on other types of resources (i.e. CPU and energy/transmit power), in a fine-grained manner, along with RBs in their RAN slicing problem. Speaking of other resources in RAN, CPU allocation is critical in virtualized RAN environments (i.e. C-RAN or vRAN) [159] where BBUs are replaced by software functions running on CPUs on vRAN [160]. Sensibly, the study in [148] analyzes the relationships among CPU and vRAN resources[6] and their influence on QoS, in the context of network slicing. The authors then formulate the CPU and RB allocation problem as a contextual-bandit problem[7] to enable customized decision making, for varied network states, at each time slot. They employ SAE to transform higher dimensional state[8] and action space into lower-dimensional one. Thereafter, AC is used for the CPU controller and DNN is used for the radio scheduler. It is noted from their experimental results on a real-world network that their scheme enables much butter throughput and network buffer state, with the same CPU resource consumption as legacy methods, while reducing the decoding error rate. On the other hand, 30% of CPU saving is achieved with their proposed framework to offer the same level of QoS that legacy methods deliver.

Furthermore, the authors in [149] exploit a DNN technique for the dynamic RB and CPU allocation of slices, while meeting the operating expenses (OPEX) cost constraints of InP. More specifically, they set lower and upper bounds on

[6]vRAN resources includes RBs, modulation and coding scheme (MCS) and transmit power.

[7]Contextual-bandit is an extension of multi-armed bandit and yet suitable for online decision problem.

[8]Higher dimensional state includes bits pending to be transmitted, mean SNR and variance SNR of each BS.

OPEX cost for specific resources (i.e. RB and CPU), while respecting SLAs between InP and slice tenants, in the DNN training processes. For the training process, per slice traffic and KPIs data is collected at cell-level from a commercial network. Based on the numerical results, the DNN technique is shown to derive solutions close to optimality, in terms of RB utilization, CPU utilization and back-haul capacity utilization.

Similarly, the authors of [127] study the RB allocation and RAN functional split to slices, while considering users' throughput, latency, and CQI. They formalize the problem as ILP and solve it using the Branch and Cut (B&C) algorithm as well as an LSTM-based approach. As their numerical results show, the LSTM-based approach is able to find near-optimal results as B&C, in terms of resource utility, throughput, latency satisfaction and RAN split deployment cost. In addition, the LSTM-based approach is observed to take 1-2ms while B&C takes 3600-3604ms.

In [150], the authors study the RB and energy allocation problems, while considering the overall system resources limitations (i.e., spectrum bandwidth, energy, queue length, etc.). Accordingly, the authors rely on a cMDP, where they integrate pre-defined constraints on these resources. To derive decisions, the SAC methods are used with on-off offline training of a SAC agent to leverage the online decision-making process. The results show that the proposed solution outperforms benchmark solutions, that do not take into account system resources constraints in the MDP formulations, in terms of QoS.

Apart from RBs, the allocation of other resources (i.e. CPU and transmit power) also has a critical impact on the end-users QoS. Accordingly, the study in [152] investigates the problem of CPU and transmit power allocation to users of multiple Service Providers (SPs), while respecting the privacy of individual SPs. In particular, the authors assume multiple SPs are competitive to each other and intend to maximize their long term payoffs. To this respect, they design the problem as an abstract stochastic game.[9] We note that in the abstract stochastic game approach, each SP evaluates the bidding values by using their local available associated users information (i.e., tasks arrival of users, queue state of users, and location of users) and abstract information of other SPs. The numerical results reveal that the DQN agent of an SP can keep the balance between CPU/transmit energy, average queue length and average resource utilization, compared to the baseline algorithms. Aside from slice-level privacy, the work in [153] takes into account device-level privacy, when deriving decisions on Transmission Power (TP) and Spreading Factor (SF)[10] in Industrial IoT (IIoT). To solve the problem, the authors consider an FL approach, where DQN agents are deployed as local agents.The FL framework is shown to be superior to centralized learning in terms of overall QoS satisfaction and energy utilization.

Mixed coarse-grained and fine-grained resource allocation: As part of RAN slicing, some papers target both coarse-grained and fine-grained. To do so, they either combine RL and heuristic approaches (see for instance [78], [125] [161], [162] [163], [164]) or combine RL and DL approaches (see for instance [89], [165]).

Specifically, the work in [161] considers both coarse-grained and fine-grained RAN resource slicing approaches. The authors introduce a framework with three main steps. Firstly, they target slicing at the coarse-grained level, where resources are partitioned over slices, based on the weight of each slice. The surplus resources are reserved to serve increases in the numbers of users later on, so as to guarantee performance isolation at all time. Accordingly, in this phase, a DDQN agent is used, allowing for dynamic adjustment of slice resources over different periods. Secondly, the authors update the BS-level resources to reflect the adjusted slice-level resources. Finally, the authors target fine-grained resource allocation. They solve the problem using a heuristic, that allows a BS to perform the RBs allocation to each UE, while ensuring QoS satisfaction and efficient resource utilization. Their simulation results show that the presented solution outperforms other baseline approaches (i.e. Q-Learning and DQN) in terms of system convergence rate and achievable reward. Similarly, in [162], the author utilizes DDPG for two-level RAN resource allocation. First, coarse-grained resource allocation to V2I and V2V slices is investigated. After that, fine-grained RBs allocation to users is done. The authors show that their DDPG outperforms DDQN, PG and AC algorithms, in terms of slice utility.

Furthermore, the authors in [163] decompose the RAN resource allocation problem into a primary problem, where slice-level resource allocations are performed, and multiple secondary problems, where slices resources are allocated to end users. To this end, the authors rely on the alternating direction method of multipliers (ADMM) to iteratively solve the master problem and DDPG to solve slave problems, with the objective of maximizing the utility of slices collectively. Without loss of generality, one may see the master problem as the coarse-grained resource allocation problem and the slave problem as the fine-grained resource allocation problem. The results reveal that the proposed approach can generate near-optimal solutions as pure ADMM and can achieve approximately 1.5 times better resource utility than the static approach.

Using the contrary approach to [163], the work in [164] first applies a DQN agent for inter-slice resource allocation (i.e. between cellular slices and D2D slices) and then ADMM for intra-D2D slice resource allocation. Again, without loss of generality, one may see inter-slice resource allocation as coarse-grained resource allocation and intra-slice resource

[9] In a stochastic game approach, to achieve the NE, every SP associated with an MVNO needs to have a global view of network dynamics, which is impractical when SPs are non-cooperative. Henceforth, they transform the stochastic game into an abstract stochastic game by only allowing to expose abstract information among SPs.

[10] Spreading Factor (SF) controls the data transmission rate of IoT devices and lower SFs mean higher data transmission rate and lower coverage.

TABLE 6. Summary table of ML-based resource allocation in network slicing. Column titles: "Input," "Decision" and "Objective" are relevant only for the papers which use combination of DL and heuristics. NA" means the required information is "Not applicable."

Scope	Ref	Focus	ML Technique	Model	State/Input	Action/Decision	Reward/Objective
RAN (Coarse-grained and fine-grained)	[161]	Maximize QoS satisfaction and resource utilization	DDQN	MDP	Resource utilization of slice, resource satisfaction and resource allocated to slice	Increase or decrease or no change of resources in percentage	Combination of QoS satisfaction and resource utilization
	[162]	Maximize the overall slice utility	DDPG	MDP	Cell utility, resource allocated to cell, QoS	Slicing ratio for uplink and downlink	Utility of slice and QoS
	[163]	Maximize the utility of slice	ADMM- DDPG	MDP	Utility function of users per slice and auxiliary/dual variables of slice	Slice resource allocation to users	Utility of slice
	[164]	Maximize the total utility of slices	DQN- ADMM	MDP	QoS satisfaction of slice, resource utilization of slice , RBs occupied by slice	Resource allocation for slice and D2D service	QoS satisfaction and resource utilization
	[89]	Minimize the long-term slices resource consumption	LSTM-A3C	MDP	Set of users per slice, resource allocated to slice and threshold of resource requirement	Slice resource allocation ratio	Combination of resource satisfaction and bonus values from reconfiguration
	[165]	Maximize QoS and resource utilization	LSTM-DDPG	MDP	Set of vehicular devices associate to slice, predicted resource demand of vehicular devices and resource reservation for vehicular devices	Scheduling action of resources for each slice	QoS satisfaction and resource efficiency
	[125]	Maximize the long term spectrum efficiency and QoS	Large-time scale: DQN Small-time scale: DDPG	MDP	Large-timescale: average user packet arrival, average delay of active users and average PDR of active users Small-time scale: user queue length, channel gain of user	Large-time scale: slice configuration, Small-time scale: PRB and power allocation to user	spectrum efficiency and QoS
	[78]	Maximize QoS of slices	ANN-TL	NA	Spatial distribution of services and users across network, services traffic, users' service request	Strategy for RAN slicing	QoS of slices
CN	[166]	Maximize the total profits of InP	Branching Dueling DQN	MDP	Demand flows of network slice, the capacity of the substrate network paths and reconfiguration cost of previous time slot	Selection of path onto which flow of network slice is mapped	Total cost of resource consumption and constraints violation
	[97]	Minimize blocking and reconfiguration penalties and average power consumption	GRU-heuristic	NA	Services data traffic trend	Resource allocation of slices	Blocking and reconfiguration penalties and average power consumption
TN	[167]	Maximize social welfare of both InP and service providers	DDPG	MDP	Resource utilization of substrate network	Increase or decrease or no change of resources unit price for each substrate nodes and links	Acceptance ratio of slice request
	[168]	Maximize the total profits of InP	DQN	MDP	Substrate network status	Turn off one network element on substrate network	Total profit of InP after turning off one network element
	[91]	Minimize average network delay	LSTM-heuristic	NA	Previous traffic matrix and link delay, demand flow	Link utilization	Average network delay
Joint RAN-CN	[169]	Maximize the total number of access users	DQN	MDP	Probability of user access to access site and ratio of users achieve E2E access to the users successfully access of the access site	Increase or decrease or no change of resources for slices	Overall access rate of the system
	[170]	Maximize resource utilization and QoE	DQN	MDP	Resource utilization and QoE satisfaction	Grouping of access units for specific slice service types	Performance of the overall system
	[36]	Maximize the total profits of slice tenants and QoS	Q-Learning	MDP	Traffic flow ratio of slice, resource demand, market price ratio and current time slot	Increase or decrease or no change of resources trading	Total profits of the tenant
	[171]	Maximize resource utility and minimum hand-off cost and outage penalty	DQN	MDP	Pair of base station and network slicing selected by users, number of users in the decision queue, serving time for users	Selection of base station and network slice pair	Cost of user being served, user hand-off cost or outage penalty cost
	[59], [163]	Maximize network throughput and minimizing hand-off cost	DQN integrated hybrid federated learning	MDP	Selected slice / base station, the available bandwidth resources of slice	Bandwidth allocation to end-user device of specific slice from specific base station	Communication efficiency and the communication cost of end-user device
	[172]	Minimize handover number, handover cost and blocking probability	Distributed Q-Learning	MDP	Bandwidth of slices	Pair of target BS/NS and handover type	Handover cost based on handover type

allocation as fine-grained resource allocation. The numerical results reveal that the presented scheme can maintain the balance between QoS and resource utilization efficiency, while ensuring performance isolation.

Notably, some contributions consider resource allocation over multiple timescales (i.e, large timescale and small timescale). As mentioned earlier, large-timescale resource allocation is referred to as coarse-grained resource allocation, and small-timescale resource allocation is referred to as fine-grained resource allocation. In this respect, the authors of [89] and [165] utilize multi-timescale in their proposed DL and RL collaborative model. They apply a DL method for large-timescale resource allocation and a RL method to perform the real-time RBs allocation to users in each slice. In both contributions, the authors design LSTM-based schemes for the large-timescale traffic forecasting and resource allocation of slices. For small-timescale resource allocation, A3C is used in [89] and DDPG is used in [165]. The experimental results suggest that the proposed collaborative model can ensure isolation among slices and is superior to other baseline RL methods (where no forecasting is considered) in terms of cumulative reward, reflecting the RB utilization.

Following the same multi-timescale idea as in [89] and [165], the work in [125] combines a conventional optimization technique and DDPG for the small-timescale RB allocation. On the other hand, DQN is used for large-timescale resource control at the base station level. To guarantee QoS and performance isolation, Guaranteed Bit Rate (GBR) and Maximum Bit Rate (MBR) constraints are integrated in their lower-level control problem. Their simulation results show that their framework outperforms the work in [89] in terms of average utility, average packet delay, and average packet drop rate. The same authors introduce TL in [78] to solve online the multi-cell RAN resource allocation problem, for vehicular networks. With TL, retraining the algorithm from scratch is not necessary, if there is a change in network states. Accordingly, the authors introduce the ideas of self-optimizing RAN slicing framework. System validation was left as future work.

2) RESOURCE ALLOCATION IN CN

Existing works on resource allocation for slices in CN are mapped to the VNF placement problem on an underlying substrate network [173].

To achieve a good level of resource efficiency, a slice might need to be reconfigured at certain intervals [174]. In general, the slice reconfiguration process includes traffic flow re-routing and VNF instances scaling on the substrate network [175]. Correspondingly, the authors of [166] investigate the intra-slice reconfiguration problems in the CN. They design a DDQN-driven framework to enhance long-term resource consumption, by encouraging the appropriate reconfiguration of slices. Their action is a set of paths onto which demand flows of the slice are mapped. Thus, their action space grows exponentially as the traffic demand increases. To deal with this, they use the action space compression method [176]. Their numerical results suggest that having an adaptive resource reconfiguration mechanism can achieve better resource efficiency, in the long run.

Similarly to [166], the authors of [97] investigate the CN slice SFC placement and routing problem as a multi-layer slice allocation problem. To this end, they consider traffic forecasting operations as well as slices reconfiguration decisions. Their main objective is to minimize reconfiguration and blocking penalties. They assume that the higher the priorities of slices are,[11] the higher the reconfiguration penalties are. GRU is used to predict the slice traffic variations for the next hour and to proactively allocate resources. Besides, their multi-layer slice allocation framework encompasses three steps: RL-based resource adjustment if predicted traffic is less than actual traffic utilization, optimal SFC placement, and routing and wavelength assignment (RWA) for nodes onto which VNFs of SFC are allocated. The algorithm validation step is left as future work.

3) RESOURCE ALLOCATION IN TN

To the best of our knowledge, none of the papers considers resource allocation in TN only. Instead, some works consider TN slicing as a subset of CN slicing or RAN slicing, to fulfill services requirements [46]. Simply speaking, TN slices act as a connectivity layer between CN domains, between RAN domains and between CN and RAN domains. In this respect, the work in [167] explores the optical data center interconnections (O-DCIs) network infrastructure to satisfy the heterogeneous slice requests from various tenants. The basic idea of O-DCIs is optical TN links are used to interconnect CN data centers to leverage cloud computing and handle the tremendous increase of data traffic in data centers [177]. In fact, network slicing in O-DCIs can ensure dynamic QoS requirements and can improve resource utilization of the CN datacenter network. However, multiple network slices allocation to O-DCIs still needs further investigation to provide cost-efficient solution for both InPs and tenants [178].

Correspondingly, the authors of [167] design the multiple slice provisioning problem as a VNE problem in the O-DCIs environment. In their work, the InP allows to tenants to solve the VNE, based on their specific slice requirements, with the objective of maximizing both the social welfare of both InP and tenants. Initially, the InP broadcasts to all tenants resource availabilities and their associated pricing framework (generated by using DDGP algorithm). The tenants then derive the optimal VNE schemes (using an ILP approach) in a distributed manner. Their pricing framework encourages the tenants not to order more resources than they need and ensures the load-balancing among tenants. In fact, ensuring the load-balancing in the network helps to reduce the blocking probability of slice requests [179]. Finally, based on VNE solutions from tenants, the InP selects the most profitable ones to allocate the corresponding resources. The evaluation results show that the proposed framework has a six time lower blocking probability than a benchmark solution, where no load-balancing is considered. DDGP is also shown to intelligently adjust the resource pricing scheme, based on the current situation of the network, to maximize both the InP and tenants' profits.

Similarly, the authors of [168] establish their research work based on the same O-DCIs reference architecture as [167] to maximize the InP's overall profits. However, in this case, the InP only relies on the DQN agent to generate the resource pricing/advertising and map the VNE requirements from MVNOs. It is observed from their results that their DQN-based framework leads to higher profit to InP in comparison to the benchmark algorithm where the InP doesn't take into account the tenants' inputs in their evaluation of resource pricing.

4) JOINT RAN AND CN RESOURCE ALLOCATION

Notably,some papers focus on joint resource allocation in RAN and CN in their network slicing framework. Accordingly, the work in [169] conducts the dynamic RBs allocation to slices in the RAN and SFC slices mapping to substrate network in the CN. Specifically, RBs are allocated in the RAN, based on the delay and rate requirements of each user. VNFs mapping to VMs then takes place in the CN. To do so, the authors rely on a DQN agent to learn the network state (covering the probability of the user being successfully attached to RAN slices and the corresponding user access rate to CN slices) and to find the optimal resource allocation policies. The simulations confirm the effectiveness of the conceptualized scheme in both static (including only stationary users) and dynamic (including mobile users) environments. Users acceptance rates of more than 98% and 97% are achieved with the proposed approach, while the baseline algorithms can achieve values of at most 94%, in the same system setting. Similarly, the work in [170] also uses a DQN agent, to find the optimal policy for the CN's computing resources allocation to slices, and a legacy optimization technique for radio resource allocation to each user in the RAN. It is clear from their results that the DQN driven framework outperforms a dynamic and a fixed resource allocation scheme, in terms of QoE and resource utilization.

[11] For example, mobile tenants have lower penalties than enterprise tenants who usually purchase more resources).

Equivalently, focusing on the resources of RAN and CN, the work in [36] seeks to maximize the long-term profits of Slice Tenants (STs) by making the best use of the assigned resources, while satisfying the QoS requirements of end-users. Having continuous user demand fluctuations, static allocation of slice resources might lead to high costs for STs. Sensibly, the proposed framework enables STs to negotiate the prices of resources with Slice Provider (SIP) and yet allows them to re-sell their surplus resources, dynamically over time. To do so, their dynamic resource trading system is designed as MDP and solved with a classical Q-Learning algorithm to achieve the long-term profits of STs. Their analysis stresses the need to have appropriate trading intervals, as short intervals may lead to high computational costs. In addition, their results show that their framework can lead to optimal QoS levels. Speaking of which, other factors affect the overall profit (i.e. handoff cost[12] and outage penalty cost).[13] Motivated by this fact, the work in [171] introduces a DQN-driven framework, reducing the unnecessary handoff cost associated with the RAN and CN resources. Their results show that their mechanism can generate better overall profits, by adjusting solutions dynamically.

All the above works are not concerned with the privacy of user data. In [59] and [180], the authors study the problem of user device association to network slices, spanning over the RAN and CN. To solve the problem, they rely on a DQN integrated Hybrid Federated Learning (FL), that exploits the benefits of both horizontal FL[14] and vertical FL[15] mechanisms. Their FL-driven framework trains the local ML model according to the local device's data and shares only the extracted features to the central model (a.k.a global model, located in the base station) to decide on the optimal selection of network slice and base station for each device. Through their simulation, their mechanism is shown to achieve better handoff cost, average network throughput, and computation efficiency, when compared to baseline approaches. Similarly, the authors in [172] attempt to solve the same handover problem, based on the distributed Q-Learning approach. However, privacy is not part of their concerns. Their scheme is shown to reduce handover frequency, cost of handover, and blocking probability to approximately 50%, as compared to conventional approaches.

5) END-TO-END NETWORK SLICING RESOURCE ALLOCATION

End-to-end (E2E) network slicing resource allocation problems are commonly seen as Virtual Network Embedding (VNE) [181] or SFC placement [182] problems, with VNFs associated to RAN, CN, and TN domains. Nevertheless, E2E resource allocation for network slicing differs from legacy VNE and SFC placement problems as it involves dealing with inter-dependent VNFs and requires as well performance isolation among slices, with distinct SLAs [183]. Additionally, E2E network slicing implies different types of resources across the RAN, CN and TN domains, making it similar to multi-resource allocation problems [184]. Besides, speaking of E2E network domains, one may note that SLA satisfaction is to be enabled across RAN, CN, and TN as well [185].

By mapping the E2E network slicing resource allocation problem to the VNE problem, in [186], the authors allocate resources to a slice over different InPs, while considering their availability in terms of link bandwidth, delay and location. Initially, they find the candidate InPs to allocate VNFs, by using a simple heuristic. Then each virtual node is associated with a set of candidate InPs. After that, the obtained set of InPs and their comprehensive features (i.e. average link bandwidth between InPs, average number of hops between InPs, average link delay, and number of VNFs associations per InP), are integrated in a CNN-based approach, to find the suitable InPs onto which VNFs are installed. As expected, the ML-based approach outperforms traditional benchmark approaches in terms of long-term revenue and computation time. Equivalently, the study in [187] designs the on-demand E2E VNEP as sMPD and solves it with DDQN, achieving a higher average revenue to the InP in the long run than DQN and greedy algorithms.

Likewise, the study in [188] and [189] rely on heuristic-driven DRL techniques (i.e PG and A3C). Specifically, the work in [188] uses a PG agent to sequentially enhance the sub-optimal solutions given by heuristic in their virtual network request (VNR) mapping (a.k.a VNE) problem, in the context of network slicing. Specifically, the outputs of the heuristic are characterized as heteograph[16] and fed to the PG, where the learning agent is designed as GCN. Their framework achieves 13-16 % improvement than baseline approaches for large-scale networks of [12], [14] nodes. On the other hand, in [189], the authors use a combined A3C and GCN approach, by which GCN extracts features from underlying networks and A3C finds the optimal policy. The actor-network of A3C is incorporated with a heuristic to maintain stability during policy training. The proposed approach outperforms the DRL-only approach, under drastic network load conditions.

In [190], the authors map the E2E slicing problem to the SFC placement problem and design an ML-based resource allocation approach that also enables SFC migration, based on end-users' mobility pattern. The proposed resource management framework is in line with ETSI-NFV standards. It encompasses an orchestration layer, a MEC/access layer and an end-users layer. The ML-based resource allocation approach is integrated into the orchestration layer. Considering both discrete and continuous action spaces, they rely

[12]Handoff cost is the cost associated to managing the mobility of users from one location to another.

[13]Outage penalty cost is the cost to pay if a user isn't served due to insufficient resources.

[14]Horizontal FL is used to aggregate the samples of end-users belongs to the same type of slice service on base stations.

[15]Vertical FL is used to aggregate the features of base stations associated with different slice services on third encrypted party side.

[16]A Heteograph is composed of a substrate graph and a VNR.

on two different DRL techniques; DQN and DDPG. The authors evaluate their framework on testbed networks. It is notable from their preliminary results that DDPG outperforms DQN in terms of stability, QoE and slice instances' downtime.

Furthermore, considering IoT use cases, the works in [191] and [192] design an ML-based adaptive resource management framework to satisfy the diverse SLAs of IoT services. Precisely, the former one proposes a network slicing automation framework that relies on LSTM to reserve resources in advance and a DQN agent to enable adaptive resource allocation, for the diverse services of IoT slices. On the other hand, the latter one introduces an ML-based multi-objective evolutionary algorithm (MOEA) to slice the IIoT network efficiently, while dealing with conflicting multiple objectives (i.e. maximizing the data throughput and minimizing the delay). It is observed from the simulation results that the solutions provide superior system performance against baseline models. Similarly, in [193], the authors formulate the online network slicing optimization problem and adopt a PPO approach to balance between SLA violations and operational costs. To say the least, their proposed solution outperforms Greedy and ILP solutions.

As mentioned earlier, E2E network slicing encompasses different types of resources, associated with different domains. Henceforth, it is crucial to have a comprehensive view of E2E network domains and corresponding resources to achieve appropriate resource allocation and enhance E2E performance. With this respect, the authors of [183] establish the 4-dimensional (4D) tensor to represent the holistic E2E network slicing model. Specifically, the 4D tensor encompasses the resource requirements of VNFs, a set of VNFs' chains, KPIs status of slicing resource management, and a concatenation of all the slices state vectors. The authors rely on DRL to solve the problem. Yet, unlike other DRL approaches previously discussed in this survey, their policy network is built using CNN to find the optimal resource adjustment policy among slices, by extracting the features of slices and their corresponding resource requirements. Their mechanism shows relatively better results over other baseline methods, in terms of SLA violation under varying network loads (120% to 200%).

Likewise, considering different types of resources across domains, the works in [194], [195], and [196] attempt to find the optimal policy for the E2E resource allocation problem for different types of slices. To represent the stochastic slice arrivals, the authors in [194] model their problem as sMDP and solve it with deep dueling Q network, while the authors in [195] and [196] model the problem as MDP and apply SPG and A2C algorithms respectively to solve the problem. All of the proposed mechanisms outperform the considered baselines, in terms of resource utilization.

While the above attempts toward E2E resource allocation consider multiple types of resources across different domains, the authors of [197] focus on E2E resource orchestration across different domains (i.e. RAN, TN, and edge). They propose a framework that encompasses decentralized resource orchestrators associated each to a domain and a central controller coordinating the network performance status. Accordingly, the authors model the resource allocation problem across the different domains as a constraint-aware MDP (a.k.a cMDP) and use a DDPG agent to solve it. According to their prototype-based evaluations,[17] their model is able to adapt to the network loads of each slice and respect the resource utilization of different domains, with a good convergence time. Besides, in comparison to baseline approaches, their framework is shown to be scalable, with approximately 3 times better system performance.

Works discussed so far on E2E resource allocation across RAN, CN and TN, do not focus on SLAs satisfaction, despite its importance. The work in [185] covers this, with the authors introducing an efficient E2E SLA decomposition framework, using ML techniques. Specifically, the authors consider three ML-based regression techniques (i.e. RF, Gradient Boosting, and Neural Network) to decompose the E2E SLAs into associated domain SLAs to create slices and assign them the required resources. Firstly, their framework checks the network capacity to accommodate the requested E2E SLA, and then it forwards the information to the classification layer to derive decisions. Besides, their framework monitors and collects the historical data of SLAs from the corresponding domains and this data is used to train the ML-based algorithms. The ultimate goal of the framework is to accurately decompose the service SLA requests into domain-level SLAs of the infrastructure layer. It is learned from their experimental results that the neural network shows higher accuracy than RF and Gradient Boosting, at the cost of lower sample efficiency.

6) LESSONS LEARNED

Unlike admission control, most of the resource allocation problems under the realm of ML in network slicing are designed as multi-objective problems. More specifically, the majority of contributions focus on the trade-off between QoS and resource efficiency by using the weighting factors approach, setting a greater factor to the more important objective. Moreover, most of the resource allocation problems in network slicing can be envisioned as sequential decision problems, naturally modeled as MDP. In particular, network slicing problems, with heterogeneous resource requirements, usually do not include the prior knowledge of transition and reward models. Thus, model-free RL frameworks are commonly used to solve them, by training an agent that derives decisions under the uncertain slice arrivals and traffic patterns, without making any assumption of the underlying network model.

Speaking of MDP formulations in resource allocation problem, in general, states are designed based on the slice

[17]The prototype consists of a Radio manager, a Transport manager and a Computing manager, configured using OpenAirInterface (OAI), OpenDayLight SDN switch and CUDA GPU platform.

TABLE 7. Summary table of ML-based resource allocation in network slicing. Column titles: "Input," "Decision" and "Objective" are relevant only for the papers which use combination of DL and heuristics. NA" means the required information is "Not applicable."

Scope	Ref	Focus	ML Technique	Model	State/Input	Action/Decision	Reward/Objective
E2E	[186]	Maximize the total profits of InP	Heuristic-CNN	NA	Number of InP and features of InP	Optimal selection of InPs	Total profit of InPs
	[187]	Maximize slice tenants revenue and minimize reconfiguration cost	DDQN	sMDP	Slice resource status, available resources of DC, changes of resource demand and additional resource requirements of slice	Reconfiguration is permitted or not	Rewarded if reconfiguration is permitted
	[188]	Maximize the revenue of InP	Relations GCN- PG	MDP	Substrate network graph and virtual network request graph	Modify the virtual nodes and link placement on substrate network graph	Revenue to cost ration of accepting virtual network requests
	[189]	Maximize slice acceptance and load balancing and minimise the overall resource consumption	A3C-heuristic	MDP	Available server resources, resource requirement of slice instance and total number of VNFs required	Placing VNFs on nodes	Positive rewards for succeed of VNFs placement, resource consumption and load balancing
	[191]	Maximize resource efficiency and robustness	LSTM- DQN	MDP	RAN, Cloud Data center and TN networks status	Generation of new slice and coordination of resources	Resource efficiency, QoS and ratio of slice failure
	[192]	Maximize data throughput and minimize the delay	ML-based MOEA	NA	Recorded services features	Map traffic to slices	Throughput and delay
	[193]	Maximize slice instance deployment and minimize deployment cost	PPO	MDP	Slice deployment demands	Slice request accepted or rejected	Positive reward if slice is accepted
	[183]	Maximize resource utilization and QoS	policy-based RL	MDP	Number of slices , different types of resources and the number of VNF nodes per slice	Increase or decrease of any VNF of each slice	Combination of SLA violation, resource wasted penalty cost and resource utilization
	[194]	Maximize profit of network provider	DDQN	sMDP	Number of slices for a given type of service	Resource allocation decision to new slice request	Purchase value of Slice Tenants
	[195]	Minimize the cost of resources and maximize the QoS	SPG	MDP	Slice resource request arrivals, buffer status	Resource allocation to slice	Resource used cost and process delay
	[196]	Maximize the overall slice utility	A2C	MDP	Slice resource request arrivals, buffer status	Resource allocation to slice	Resource used cost and process delay
	[197]	Maximize the overall system performance	DDPG	cMDP	Allocated and demanded resources for node and links of slice	Resource allocation to slice	Performance of the traffic flows
	[185]	Minimize E2E delay	Random Forest, Gradient Boosting and Neural Network	NA	E2E slice creation request	Decomposition of network domain SLAs	E2E delay

status (i.e. resource utilization, QoS satisfaction, the number of users/slice, the demand traffic flows, and so forth), actions are mainly denoted as the resource allocation actions (i.e. increasing/decreasing resources or keeping the same resources) and rewards are typically formulated to meet the main objectives of the problem. For instance, if the objective of a problem is the maximization of the long-term spectrum efficiency and QoE, the reward can be a weighted combination of spectrum efficiency and QoE. As one may notice, the number of users, slices, or demand of traffic flows have been increasing rapidly day over day. Accordingly, the curse of dimensionality issue of state-action pair arises in the context of resource allocation. To overcome this, some papers adopt certain methods (i.e. reward clipping, SAE) in their RL framework to improve sample efficiency and filter out unnecessary state-action pairs. Regardless of being well-known techniques, ML-based algorithms in network slicing resource allocation have certain pitfalls in terms of sample efficiency, training time, and algorithm complexity, henceforth calling for further investigations. All the above contributions are summarized in Table 5, Table 6 and Table 7.

V. OPEN RESEARCH CHALLENGES AND OPPORTUNITIES

While this survey underlines the recent achievements of several ML-based methods in network slicing problems, there are still some open challenges in incorporating ML theory and algorithms for practical network slicing deployment. In this section, we identify the open challenges of integrating ML solutions in network slicing. From this perspective, admission control and resource allocation generally fall into one category, since the same ML techniques can be applied for both. To this end, we discuss and underline some of the critical open challenges and research gaps, mainly twofold: (i) open challenges particularly related to deep learning in traffic forecasting, and (ii) open challenges particularly focused on reinforcement learning in network slicing admission control and resource allocation. While identifying these open challenges, we also discuss potential solutions for these problems.

A. ML IN TRAFFIC FORECASTING FOR NETWORK SLICING

After extensively studying the related articles on forecasting in network slicing, we are able to distinguish some specific challenges, such as forecasting slice-level traffic while accounting for the privacy of MVNOs, end-user level traffic forecasting, 1 ms forecasting granularity, and the trade-off between computing complexity and accuracy.

In most of the existing state-of-the-art works, the forecasting function resides in the central controller of the network and is managed by the InP. However, sometimes, due to privacy concerns, MVNOs cannot share their slice traffic information, which is entirely related to the behavior of their users, with the InP. Consequently, this calls for an entirely new forecasting framework, which allows MVNOs to train their algorithms locally and only share non-sensitive information with the central controller (managed by InP), to be used for resource reservation during the next time interval (or time window). It is worth stressing that privacy-preserving network slice traffic forecasting still has room to be improved. In this regard, one potential solution would be exploring the concept of federated learning (a.k.a decentralized learning) [198]. Technically, FL allows multiple actors to train and control their models locally, without exchanging any critical data with the central controller. That being said, FL-driven solutions shall be further investigated to deal with multi-stakeholders network slicing environments which exhibit diverse security requirements.

It is also important to point out that forecasting is done only for the aggregated slice-level or RAN-level traffic in the existing state-of-the-art. Hence, forecasting of end-user traffic is still a missing piece of the network slicing problem. Knowing the behavior of individual users in advance is certainly beneficial to fulfill the individual user QoS requirements. In this respect, giving authority to the MVNOs to forecast the traffic closer to the user, chances are they have better proficiency for end-user level traffic forecasting. If reasonable accuracy is obtained, one might want to feed this kind of end-user level forecast data into end-user admission control solutions or in fine-grained resource allocation systems where end-user level QoS satisfaction is explicitly considered.

In our literature study, we encountered some achievements of incorporating forecasting into the resource allocation function. However, one should be aware of the high dynamicity of traffic requirements in mobile network environments, where traffic demand often varies in the order of milliseconds, a trend which is likely to increase in the future. Henceforth, dynamic resource allocation granularity in the RAN is conducted at the level of RB, with TTI of 1 ms [44], shown to achieve better resource efficiency than static RB allocation [125]. Correspondingly, an anticipatory resource allocation approach requires to have forecast data at a millisecond time interval granularity. It is notable that all the traffic forecasting solutions in existing network slicing problems rely on a time interval of seconds or minutes, which does not seem adapted to these problems. This calls for further investigation with millisecond-level time intervals. In such cases, there are obvious challenges in obtaining datasets with such level of granularity. If one has system capabilities that can capture the desired level of granularity, the forecasting function then needs to be trained using a good amount of such data to get to acceptable accuracy values. Besides, it is worth exploiting current ML solutions, tested with data at a minute or hour granularity, which is prone to obtain reasonable results at the millisecond level.

Another open topic corresponds to the trade-off between computing complexity and accuracy in the training of DL algorithms used for network slicing. It is important to explicitly investigate the algorithm complexity and determine the level of computing complexity required to attain the desired level of accuracy in the slice traffic forecasting. In this case, one should have a vivid understanding of the requirements and objectives of the users and system, to be able to choose the right forecasting technique corresponding to specific requirements. From our knowledge, none of the existing papers explicitly justify their choice for one specific forecasting technique over other by considering both accuracy and complexity perspectives. It is clear that forecasting techniques shall be selected based on a compromise between complexity and accuracy. Specifically, if a system has no limit to computing power, one may want to deploy a Transformer Model [199] to get better accuracy than LSTM [200]. On the other hand, GRU will be selected against LSTM in a system with limited computation time [95].

B. ML IN ADMISSION CONTROL AND RESOURCE ALLOCATION

In our literature review, we identified some unique challenges in the context of admission control and resource allocation problems for network slicing, such as satisfying different slice requirements in one iteration, accomplishing more than one objective simultaneously, handling high computing complexity due to an exponential number of state-action pairs and energy overheads incurred due to massive communication of collaborative decentralized ML models in 6G network slicing.

Indeed, different network slice types (i.e., eMBB, URLLC and mMTC) have remarkably diverse requirements. Nevertheless, the majority of state-of-the-art RL solutions in network slicing admission control and resource allocation rely on a single agent to deal with the heterogeneous slice requirements and apply that same single agent with the same reward function to all the slices consecutively. From our review, only two papers ([89], [147]) use multiple agents. However, since the nature of slices is heterogeneous, it is impractical to use one single agent, trained only for one specific purpose, in the real-world diversified slice deployments. Here, potential solutions are to explore the concept of multi-agent RL (MARL) [201]. In essence, MARL is where multiple agents are interacting with the common network environment to find the optimal policy based on their associated reward functions. Based on the design of the reward scheme, one may have a system where multiple agents are working cooperatively or competitively, or even a mixed cooperative and competitive mode.

In general, there are inherent trade-offs between multiple objectives in the admission control and resource allocation problems. For example, there is a trade-off between user admission probability and QoS satisfaction, a trade-off between network reliability and resource efficiency, and so forth. In all of the existing RL solutions, those trade-offs are handled in the reward function, by defining a weighting factor between two objectives. Nonetheless, to do so, the weighting factor must be predefined, and it is hard to assure that the applied weighting factor is an optimal one for the given scenario. In this respect, methods such as multi-objective RL (MORL) [202], known to solve problems with conflicting objectives, might come in handy. In MORL, the reward function is designed as a vector instead of a scalar value. MORL returns a reward vector for the respective individual objectives, rather than returning a scalar reward value.

The curse of dimensionality is an inherent and ongoing issue of RL in network slicing admission control and resource allocation problems, due to an enormous number of state-action pairs. Notably, the number of base stations, slices, and end-user devices is growing continuously. We came across very few attempts to address this problem in the existing literature, the main example being the application of sparse autoencoders [148] to reduce the number of dimension in the state-action space. More efforts are needed to further investigate such state-action reduction tools, for example deep autoencoders [203], latent variable models [204], etc. Also, the learning agent utilizes the random policy in the initial steps of all the existing RL solutions applied to network slicing problems, thereby leading to longer convergence time. On this matter, one possibility is exploiting the idea of imitation learning [205] to enable the learning agent to start with a relatively adapted policy, which reduces the convergence time, instead of initializing it with a risky random policy.

Indeed, sample efficiency is also one of the key ongoing challenges to enhance the RL agent training process in network slicing problems. In a real-world network slicing environment, we rarely get the desired level of data to train an agent, which requires to carefully design frameworks which can be trained efficiently with the available data. Noticeably, the number of studies focused on sample efficiency is considerably low in our review, and this is an area where additional efforts are expected. To this end, reward shaping and TL are the most common approaches in the literature to improve sample efficiency and leverage the learning process of RL agents. Moreover, propitious solutions in this sense are based on meta-learning [206] and hierarchical DNN (HiDeNN) [207] frameworks. Note that HiDeNN can be built with any type of neural network (i.e., DNN, RNN, or CNN) [208]. Technically, meta-learning enables the agent to improve its learning process with a minimalist amount of samples by initializing the training with optimized hyper-parameters from prior knowledge. On the other hand, recent frameworks such as HiDeNN can also enhance the learning efficiency with less amount of data, with the help of TL [207]. An interesting property is that agents in HiDeNN can learn from incremental changes in data applied to a previously trained model [209]. For instance, in the case of network slicing, this can represent new user/slice requests, which arrive incrementally over time.

Last but not least, energy efficiency is one of the key performance indicators for the sustainability of multilayer network slicing. Regardless of its benefits, a centralized ML model exacerbates the energy consumption of the overall network. Compared to the energy required for local computations, the energy required for raw data transmission is much higher [210]. Hence, it is logical to put forward advanced decentralized architectures, such as MARL and federated DRL to avoid unnecessary raw data transmission to the central nodes and exploit the local computing power. Since 6G network is envisioned as a multilayer heterogeneous network [211], thousands of ML agents might be involved in the collaborative decentralized ML scheme. This would lead to higher energy consumption due to a large number of communication (i.e. sharing model updates) between ML agents in the regime of collaborative learning. To diminish energy overheads, one might want to develop a federate framework that selects the optimal number of participants in a more intelligent manner in the collaborative training process while achieving the desired accuracy. That said, one promising solution is the MAB-based FL approach [212], in which the MAB agent selects the most auspicious participants which can better leverage the performance of the overall model.

VI. CONCLUSION

In this survey, we focus on the applications of ML techniques in network slicing. First, we present background information on network slicing. We then provide an overview of some common ML techniques, used in network slicing. After that, we review the literature on the topic. In particular, we group contributions into three categories: traffic forecasting, admission control and resource allocation. For each category, we highlight lessons learned. Finally, we discuss some open challenges and hint to potential solutions that can be considered.

REFERENCES

[1] Ericsson. (Jun. 2021). *Ericsson Mobility Report*. [Online]. Available: https://www.ericsson.com/en/mobility-report/reports/june-2021
[2] *Network Slicing: Top 10 use Cases to Target*, Ericsson, Stockholm, Sweden, 2021.
[3] X. Foukas, G. Patounas, A. Elmokashfi, and M. K. Marina, "Network slicing in 5G: Survey and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 94–100, May 2017.
[4] *IMT Vision Framework and Overall Objectives of the Future Development of IMT for 2020 and Beyond*, document ITU-R M.2083-0, 2015.
[5] A. Ksentini and N. Nikaein, "Toward Enforcing Network Slicing on RAN: Flexibility and Resources Abstraction," *IEEE Commun. Mag.*, vol. 55, no. 6, pp. 102–108, Jun. 2017.
[6] H. Strothers, R. Lewis, R. Hood, and R. Butcher, "NGMN 5G white paper," *J. Nat. Med. Assoc.*, vol. 96, no. 2, p. 124, 2004. [Online]. Available: https://www.ngmn.org/uploads/media/NGMN_5G_White_Paper_V1_0_01.pdf
[7] *GSMA Network Slicing*, UC Requirements, Groupe Speciale Mobile Assoc., London, U.K., Apr. 2018.
[8] T-Specification, "Study on architecture for next generation system, release 14," 3GPP, Sophia Antipolis, France, Tech. Rep. TR 23.799, 2016.
[9] Ericsson and A. D. Little. (Jan. 2021). *Network Slicing: A go-to-Market Guide to Capture the High Revenue Potential*. [Online]. Available: https://www.ericsson.com/assets/local/digital-services/network-slicing/network-slicing-value-potential.pdf?_ga=2.108774145.224934064.1636121875-555001856.1630593225#page=8&zoom=auto,-274,278
[10] M. Chahbar, G. Diaz, A. Dandoush, C. Christophe, and K. Ghoumid, "A comprehensive survey on the E2E 5G network slicing model," *IEEE Trans. Netw. Service Manag.*, vol. 4537, no. 1, pp. 49–62, Mar. 2020.
[11] S. D. A. Shah, M. A. Gregory, and S. Li, "Cloud-native network slicing using software defined networking based multi-access edge computing: A survey," *IEEE Access*, vol. 9, pp. 10903–10924, 2021.
[12] F. Debbabi, R. Jmal, L. C. Fourati, and A. Ksentini, "Algorithmics and modeling aspects of network slicing in 5G and beyonds network: Survey," *IEEE Access*, vol. 8, pp. 162748–162762, 2020.
[13] M. O. Ojijo and O. E. Falowo, "A survey on slice admission control strategies and optimization schemes in 5G network," *IEEE Access*, vol. 8, pp. 14977–14990, 2020.
[14] S. Wijethilaka and M. Liyanage, "Survey on network slicing for Internet of Things realization in 5G networks," *IEEE Commun. Surveys Tuts.*, vol. 23, no. 2, pp. 957–994, 2nd Quart., 2021.
[15] L. U. Khan, I. Yaqoob, N. H. Tran, Z. Han, and C. S. Hong, "Network slicing: Recent advances, taxonomy, requirements, and open research challenges," *IEEE Access*, vol. 8, pp. 36009–36028, 2020.
[16] R. Su, D. Zhang, R. Venkatesan, Z. Gong, C. Li, F. Ding, F. Jiang, and Z. Zhu, "Resource allocation for network slicing in 5G telecommunication networks: A survey of principles and models," *IEEE Netw.*, vol. 33, no. 6, pp. 172–179, Nov./Dec. 2019.
[17] J. A. Hurtado Sánchez, K. Casilimas, and O. M. Caicedo Rendon, "Deep reinforcement learning for resource management on network slicing: A survey," *Sensors*, vol. 22, no. 8, p. 3031, Apr. 2022.
[18] C. Ssengonzi, O. P. Kogeda, and T. O. Olwal, "A survey of deep reinforcement learning application in 5G and beyond network slicing and virtualization," *Array*, vol. 14, Jul. 2022, Art. no. 100142, doi: 10.1016/j.array.2022.100142.

[19] C. Telecom, Y. Bai, D. Sun, J. S. Huawei, L. Licciardi, H. Li, L. Wang, A. Artigiani, D. Sabella, H. Wang, C. Maitre, P. Telecom, F. Fontes, Y. Wang, S. Cadzow, and C. Corbi, *Autonomous Networks, Supporting Tomorrow's ICT Business 2*. Accessed: Oct. 2020. [Online]. Available: https://www.etsi.org

[20] J. Xie, Z. Song, Y. Li, Y. Zhang, H. Yu, J. Zhan, Z. Ma, Y. Qiao, J. Zhang, and J. Guo, "A survey on machine learning-based mobile big data analysis: Challenges and applications," *Wireless Commun. Mobile Comput.*, vol. 2018, pp. 1–19, Aug. 2018.

[21] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2224–2287, 3rd Quart., 2019.

[22] *Description of Network Slicing Concept*, 5G Project Requirements & Archit. Work Stream E2E Archit., NGMN, Germany, vol. 1, Jan. 2016, pp. 1–7. [Online]. Available: https://www.ngmn.org/fileadmin/user_upload/161010_NGMN_Network_Slicing_framework_v1.0.8.pdf

[23] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," *Comput. Netw.*, vol. 167, Feb. 2020, Art. no. 106984.

[24] *Next Generation Protocols (NGP); E2E Network Slicing Reference Framework and Information Model*, I-Models, Eur. Telecommun. Standards Inst., Sophia Antipolis, France, vol. 1, 2018, pp. 1–32.

[25] T. Shimojo, M. R. Sama, A. Khan, and S. Iwashina, "Cost-efficient method for managing network slices in a multi-service 5G core network," in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, May 2017, pp. 1121–1126.

[26] V. P. Kafle, Y. Fukushima, P. Martinez-Julia, and T. Miyazawa, "Consideration on automation of 5G network slicing with machine learning," in *Proc. ITU Kaleidoscope, Mach. Learn. for a 5G Future (ITU K)*, Nov. 2018, pp. 1–8.

[27] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, K. Samdanis, and X. Costa-Perez, "Optimising 5G infrastructure markets: The business of network slicing," in *Proc. IEEE Conf. Comput. Commun.*, May 2017, pp. 1–9.

[28] L. Guijarro, J. R. Vidal, and V. Pla, "Competition between tenants with strategic resource allocation in 5G network slicing," 2019, *arXiv:1912.04764*.

[29] V. Sciancalepore, M. Di Renzo, and X. Costa-Perez, "STORNS: Stochastic radio access network slicing," 2019, *arXiv:1901.05336*.

[30] M. Vincenzi, E. Lopez-Aguilera, and E. Garcia-Villegas, "Maximizing infrastructure providers' revenue through network slicing in 5G," *IEEE Access*, vol. 7, pp. 128283–128297, 2019.

[31] Y. K. Tun, N. H. Tran, D. T. Ngo, S. R. Pandey, Z. Han, and C. S. Hong, "Wireless network slicing: Generalized Kelly mechanism-based resource allocation," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 8, pp. 1794–1807, Aug. 2019.

[32] P. Munoz, O. Adamuz-Hinojosa, J. Navarro-Ortiz, O. Sallent, and J. Perez-Romero, "Radio access network slicing strategies at spectrum planning level in 5G and beyond," *IEEE Access*, vol. 8, pp. 79604–79618, 2020.

[33] S. K. A. Kumar, R. Stewart, D. Crawford, and S. Chaudhari, "Business model for rural connectivity using multi-tenancy 5G network slicing," in *Proc. IEEE 17th Int. Conf. Smart Communities, Improving Quality Life Using ICT, IoT AI (HONET)*, Dec. 2020, pp. 182–188.

[34] H. Zhao, S. Deng, Z. Liu, Z. Xiang, J. Yin, S. Dustdar, and A. Zomaya, "DPoS: Decentralized, privacy-preserving, and low-complexity online slicing for multi-tenant networks," *IEEE Trans. Mobile Comput.*, vol. 1233, no. 12, pp. 1–14, Dec. 2021.

[35] S. Kuklinski, L. Tomaszewski, K. Kozlowski, and S. Pietrzyk, "Business models of network slicing," in *Proc. 9th Int. Conf. Netw. Future (NOF)*, no. 723172, Nov. 2018, pp. 39–43.

[36] Y. Kim, S. Kim, and H. Lim, "Reinforcement learning based resource management for network slicing," *Appl. Sci.*, vol. 9, no. 11, p. 2361, 2019.

[37] X. Chen, Z. Zhao, C. Wu, M. Bennis, H. Liu, Y. Ji, and H. Zhang, "Multi-tenant cross-slice resource orchestration: A deep reinforcement learning approach," 2018, *arXiv:1807.09350*.

[38] M. Toscano, F. Grunwald, M. Richart, J. Baliosian, E. Grampín, and A. Castro, "Machine learning aided network slicing," in *Proc. Int. Conf. Transparent Opt. Netw.*, Jul. 2019, pp. 1–4.

[39] J.-B. Monteil, J. Hribar, P. Barnard, Y. Li, and L. A. DaSilva, "Resource reservation within sliced 5G networks: A cost-reduction strategy for service providers," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, Jun. 2020, pp. 1–6.

[40] H. Chergui and C. Verikoukis, "Offline SLA-constrained deep learning for 5G networks reliable and dynamic end-to-end slicing," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 350–360, Feb. 2020.

[41] (Jun. 2021). *E2E Network Slicing Architecture Version 1.0 03*. [Online]. Available: https://www.gsma.com/newsroom/wp-content/uploads//NG.127-v1.0-2.pdf

[42] T-Specification, "LTE; evolved universal terrestrial radio access (E-UTRA); radio resource control (RRC); protocol specification (3GPP TS 36.331 version 8.3.0 Release 8)," ETSI, Sophia Antipolis, France, Tech. Rep. TS 136 331, V8.3.0, 2008.

[43] X. Li, R. Ni, J. Chen, Y. Lyu, Z. Rong, and R. Du, "End-to-end network slicing in radio access network, transport network and core network domains," *IEEE Access*, vol. 8, pp. 29525–29537, 2020.

[44] T-Specification, "Technical specification group radio access network; study on new radio access technology: Radio access architecture and interfaces (3GPP TR 38.801 version 14.0.0 Release 14)," 3GPP, Valbonne, France, Tech. Rep. TR 38.801, V14.0.0, 2017.

[45] ETSI, "5G; System architecture for the 5G System (5GS) (3GPP TS 23.501 version 15.9.0 Release 15)," ETSI, Sophia Antipolis, France, Tech. Rep. TS 123 501, V15.9.0, 2020, pp. 385–393. [Online]. Available: https://www.etsi.org/deliver/etsi_ts/123500_123599/123501/15.09.00_60/ts_123501v150900p.pdf

[46] (2020). *IETF Definition of Transport Slice Draft-NSDT-Teas-Transport-Slice-Definition-04*. [Online]. Available: https://datatracker.ietf.org/doc/html/draft-nsdt-teas-transport-slice-definition-04#page-11

[47] (2020). *Broadband Forum Motivates 5G Transport Network Architecture and Requirements in New Whitepaper*. [Online]. Available: https://www.broadband-forum.org

[48] Q. Ye, J. Li, K. Qu, W. Zhuang, X. S. Shen, and X. Li, "End-to-end quality of service in 5G networks: Examining the effectiveness of a network slicing framework," *IEEE Trans. Veh. Technol.*, vol. 13, no. 2, pp. 65–74, Jun. 2018.

[49] X. Foukas, M. K. Marina, and K. Kontovasilis, "Orion: RAN slicing for a flexible and cost-effective multi-service mobile network architecture," in *Proc. 23rd Annu. Int. Conf. Mobile Comput. Netw.*, vol. F1312, Oct. 2017, pp. 127–140.

[50] *5G NR Logical Architecture and its Functional Splits*, IW Suite, Parallel Wireless, Nashua, U.K., 2021, no. 603.

[51] I. Afolabi, T. Taleb, K. Samdanis, A. Ksentini, and H. Flinck, "Network slicing and softwarization: A survey on principles, enabling technologies, and solutions," *IEEE Commun. Surveys Tuts.*, vol. 20, no. 3, pp. 2429–2453, 3rd Quart., 2018.

[52] I. Afolabi, M. Bagaa, T. Taleb, and H. Flinck, "End-to-end network slicing enabled through network function virtualization," in *Proc. IEEE Conf. Standards for Commun. Netw. (CSCN)*, Sep. 2017, pp. 30–35.

[53] O. Sallent, J. Pérez-Romero, R. Ferrús, and R. Agustí, "On radio access network slicing from a radio resource management perspective," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 166–174, Oct. 2017.

[54] A. Gudipati, A. Perry, L. E. Li, and S. Katti, "SoftRAN: Software defined radio access network," in *Proc. 2nd ACM SIGCOMM Workshop Hot Topics Softw. Defined Netw. (HotSDN)*, Hong Kong. New York, NY, USA: Association for Computing Machinery, 2013, pp. 25–30, doi: 10.1145/2491185.2491207.

[55] X. Foukas, N. Nikaein, M. M. Kassem, M. K. Marina, and K. Kontovasilis, "FlexRAN: A flexible and programmable platform for software-defined radio access networks," in *Proc. 12th Int. Conf. Emerg. Netw. Experiments Technol.*, 2016, pp. 427–441.

[56] T-Specification, "3rd generation partnership project; technical specification group services and system aspects; system architecture for the 5G system (5GS); stage 2 (3GPP TS 23.501 version 17.0.0 Release 17)," Tech. Rep. TS 23.501, V17.0.0, 3rd Generation Partnership Project, Sophia Antipolis, France, 2021.

[57] M. Ivezic. *Introduction to 5G Core Service-Based Architecture (SBA) Components*. Accessed: Aug. 16, 2020. [Online]. Available: https://5g.security/5g-technology/5g-core-sba-components-architecture/

[58] M. Mohtasim Sajjad, C. J. Bernardos, D. Jayalath, and Y.-C. Tian, "Inter-slice mobility management in 5G: Motivations, standard principles, challenges and research directions," 2020, *arXiv:2003.11343*.

[59] Y.-J. Liu, G. Feng, Y. Sun, S. Qin, and Y.-C. Liang, "Device association for RAN slicing based on hybrid federated deep reinforcement learning," *IEEE Trans. Veh. Technol.*, vol. 69, no. 12, pp. 15731–15745, Dec. 2020.

[60] Y.-I. Choi and N. Park, "Slice architecture for 5G core network," in *Proc. 9th Int. Conf. Ubiquitous Future Netw. (ICUFN)*, Jul. 2017, pp. 571–575.

[61] T. V. Doan, G. T. Nguyen, H. Salah, S. Pandi, M. Jarschel, R. Pries, and F. H. P. Fitzek, ''Containers vs virtual machines: Choosing the right virtualization technology for mobile edge cloud,'' in *Proc. IEEE 2nd 5G World Forum (5GWF)*, Sep. 2019, pp. 46–52.
[62] (2020). *Nokia White Paper Transport Slicing in end-to-end 5G Networks*. [Online]. Available: https://pages.nokia.com/T004QE-Transport-slicing-in-end-to-end-5G-networks
[63] M. R. Raza, M. Fiorani, A. Rostami, P. Öhlen, L. Wosinska, and P. Monti, ''Dynamic slicing approach for multi-tenant 5G transport networks [invited],'' *J. Opt. Commun. Netw.*, vol. 10, no. 1, p. A77, 2018.
[64] F. Tonini, C. Natalino, M. Furdek, C. Raffaelli, and P. Monti, ''Network slicing automation: Challenges and benefits,'' in *Proc. Int. Conf. Opt. Netw. Design Model. (ONDM)*, May 2020, pp. 1–6.
[65] M. Hadi, M. R. Pakravan, and E. Agrell, ''Dynamic resource allocation in metro elastic optical networks using Lyapunov drift optimization,'' *J. Opt. Commun. Netw.*, vol. 11, no. 6, pp. 250–259, 2019.
[66] R. Casellas, A. Giorgetti, R. Morro, R. Martinez, R. Vilalta, and R. Muñoz, ''Virtualization of disaggregated optical networks with open data models in support of network slicing,'' *J. Opt. Commun. Netw.*, vol. 12, no. 2, p. A144, 2020.
[67] V. Curri and D. Calonico, ''Multi-band and multi-service open optical networks: Applications and perspectives,'' in *Proc. 22nd Int. Conf. Transparent Opt. Netw. (ICTON)*, Jul. 2020, pp. 1–4.
[68] R. Alvizu, G. Maier, N. Kukreja, A. Pattavina, R. Morro, A. Capello, and C. Cavazzoni, ''Comprehensive survey on T-SDN: Software-defined networking for transport networks,'' *IEEE Commun. Surveys Tuts.*, vol. 19, no. 4, pp. 2232–2283, 4th Quart. 2017.
[69] A. de la Oliva, X. Li, X. Costa-Perez, C. J. Bernardos, P. Bertin, P. Iovanna, T. Deiss, J. Mangues, A. Mourad, C. Casetti, J. E. Gonzalez, and A. Azcorra, ''5G-TRANSFORMER: Slicing and orchestrating transport networks for industry verticals,'' *IEEE Commun. Mag.*, vol. 56, no. 8, pp. 78–84, Aug. 2018.
[70] (2018). *Framework for Abstraction and Control of TE Networks (ACTN)*. [Online]. Available: https://datatracker.ietf.org/doc/html/rfc8453
[71] M. Ersue and ETSI NFV MANO WG, ''ETSI NFV management and orchestration—An overview virtualization as a paradigm,'' in *Proc. IETF*, 2013, pp. 1–14.
[72] A. M. Medhat, T. Taleb, A. Elmangoush, G. A. Carella, S. Covaci, and T. Magedanz, ''Service function chaining in next generation networks: State of the art and research challenges,'' *IEEE Commun. Mag.*, vol. 55, no. 2, pp. 216–223, Feb. 2017.
[73] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, ''A survey on software-defined networking,' *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 27–51, 1st Quart., 2015.
[74] A. Lenk, M. Klems, J. Nimis, S. Tai, and T. Sandholm, ''What's inside the cloud? An architectural map of the cloud landscape,'' in *Proc. ICSE Workshop Softw. Eng. Challenges Cloud Comput.*, 2009, pp. 23–31.
[75] A. Zappone, M. Di Renzo, and M. Debbah, ''Wireless networks design in the era of deep learning: Model-based, AI-based, or both?'' *IEEE Trans. Commun.*, vol. 67, no. 10, pp. 7331–7376, Oct. 2019.
[76] N. Bui and J. Widmer, ''Data-driven evaluation of anticipatory networking in LTE networks,'' *IEEE Trans. Mobile Comput.*, vol. 17, no. 10, pp. 2252–2265, Oct. 2018.
[77] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia, and A. Banchs, ''Mobile traffic forecasting for maximizing 5G network slicing resource utilization,'' in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, May 2017, pp. 1–4.
[78] J. Mei, X. Wang, and K. Zheng, ''An intelligent self-sustained RAN slicing framework for diverse service provisioning in 5G-beyond and 6G networks,'' *Intell. Converged Netw.*, vol. 1, pp. 281–294, Dec. 2020.
[79] Z. Xiong, Y. Zhang, and D. Niyato, ''Deep reinforcement learning for mobile 5G and beyond: Fundamentals, applications, and challenges,'' *IEEE Veh. Technol. Mag.*, vol. 14, no. 2, pp. 44–52, Jun. 2019.
[80] T. V. K. Buyakar, H. Agarwal, B. R. Tamma, and A. A. Franklin, ''Resource allocation with admission control for GBR and delay QoS in 5G network slices,'' in *Proc. Int. Conf. Commun. Syst. Netw. (COMSNETS)*, 2020, pp. 213–220.
[81] A. T. Ajibare and O. E. Falowo, ''Resource allocation and admission control strategy for 5G networks using slices and users priorities,'' in *Proc. IEEE AFRICON Conf.*, Sep. 2019, pp. 1–6.
[82] C. Marquez, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, ''Resource sharing efficiency in network slicing,'' *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 3, pp. 909–923, Sep. 2019.
[83] T. Jirsik, S. Trcka, and P. Celeda, ''Quality of service forecasting with LSTM neural network,'' in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manag. (IM)*, Apr. 2019, pp. 251–260.
[84] S. Siami-Namini, N. Tavakoli, and A. S. Namin, ''A comparison of ARIMA and LSTM in forecasting time series,'' in *Proc. 17th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2018, pp. 1394–1401.
[85] C.-W. Huang, C.-T. Chiang, and Q. Li, ''A study of deep learning networks on mobile traffic forecasting,'' in *Proc. IEEE 28th Annu. Int. Symp. Pers., Indoor, Mobile Radio Commun. (PIMRC)*, Oct. 2017, pp. 1–6.
[86] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, ''DeepCog: Optimizing resource provisioning in network slicing with AI-based capacity forecasting,'' *IEEE J. Sel. Areas Commun.*, vol. 38, no. 2, pp. 361–376, Feb. 2020.
[87] D. Bega, M. Gramaglia, A. Garcia-Saavedra, M. Fiore, A. Banchs, and X. Costa-Perez, ''Network slicing meets artificial intelligence: An AI-based framework for slice management,'' *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 32–38, Jun. 2020.
[88] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, ''AZTEC: Anticipatory capacity allocation for zero-touch network slicing,'' in *Proc. IEEE Conf. Comput. Commun.*, Jul. 2020, pp. 794–803.
[89] M. Yan, G. Feng, J. Zhou, Y. Sun, and Y. Liang, ''Intelligent resource scheduling for 5G radio access network slicing,'' *IEEE Trans. Veh. Technol.*, vol. 68, no. 8, pp. 7691–7703, Aug. 2019.
[90] S. K. Tayyaba, H. A. Khattak, A. Almogren, M. A. Shah, I. U. Din, I. Alkhalifa, and M. Guizani, ''5G vehicular network resource management for improving radio access through machine learning,'' *IEEE Access*, vol. 8, pp. 6792–6800, 2020.
[91] E. H. Bouzidi, A. Outtagarts, A. Hebbar, R. Langar, and R. Boutaba, ''Online based learning for predictive end-to-end network slicing in 5G networks,'' in *Proc. IEEE Int. Conf. Commun.*, Jun. 2020, pp. 1–7.
[92] R. Li, C. Wang, Z. Zhao, R. Guo, and H. Zhang, ''The LSTM-based advantage actor-critic learning for resource management in network slicing with user mobility,'' *IEEE Commun. Lett.*, vol. 24, no. 9, pp. 2005–2009, Sep. 2020.
[93] C. Zhang, M. Fiore, and P. Patras, ''Multi-service mobile traffic forecasting via convolutional long short-term memories,'' 2019, *arXiv:1905.09771*.
[94] Y. Cui, X. Huang, D. Wu, and H. Zheng, ''Machine learning-based resource allocation strategy for network slicing in vehicular networks,'' *Wireless Commun. Mobile Comput.*, vol. 2020, pp. 1–10, Nov. 2020.
[95] S. Yang, X. Yu, and Y. Zhou, ''LSTM and GRU neural network performance comparison study: Taking yelp review dataset as an example,'' in *Proc. Int. Workshop Electron. Commun. Artif. Intell. (IWECAI)*, Jun. 2020, pp. 98–101.
[96] H. Chergui and C. Verikoukis, ''Big data for 5G intelligent network slicing management,'' *IEEE Netw.*, vol. 34, no. 4, pp. 56–61, Aug. 2020.
[97] R. Alvizu, S. Troia, G. Maier, and A. Pattavina, ''Machine-Learning-Based Prediction and Optimization of Mobile Metro-Core Networks,'' in *Proc. IEEE Photon. Soc. Summer Topicals Meeting Ser. (SUM)*, no. 1, Jul. 2018, pp. 155–156.
[98] H. P. Phyu, D. Naboulsi, R. Stanica, H. P. Phyu, D. Naboulsi, R. Stanica, M. Traffic, N. Slices, and H. P. Phyu, ''Mobile traffic forecasting for network slices: A federated-learning approach,'' in *Proc. IEEE 33rd Annu. Int. Symp. Pers., Indoor Mobile Radio Commun. (PIMRC)*, Sep. 2022, pp. 745–751.
[99] C. Song, M. Zhang, X. Huang, Y. Zhan, D. Wang, M. Liu, and Y. Rong, ''Machine learning enabling traffic-aware dynamic slicing for 5G optical transport networks,'' in *Proc. Conf. Lasers Electro-Optics*, 2018, pp. 1–2.
[100] B. Brik and A. Ksentini, ''On predicting service-oriented network slices performances in 5G: A federated learning approach,'' in *Proc. Conf. Local Comput. Netw. (LCN)*, Nov. 2020, pp. 164–171.
[101] A. Thantharate and C. Beard, ''ADAPTIVE6G: Adaptive resource management for network slicing architectures in current 5G and future 6G systems,'' *J. Netw. Syst. Manage.*, vol. 31, no. 1, pp. 1–24, Jan. 2023, doi: 10.1007/s10922-022-09693-1.
[102] M. Jiang, M. Condoluci, and T. Mahmoodi, ''Network slicing management & prioritization in 5G mobile systems,'' in *Proc. Eur. Wireless Conf.*, 2016, pp. 197–202.
[103] B. Han, D. Feng, and H. D. Schotten, ''A Markov model of slice admission control,'' *IEEE Networking Letters*, vol. 1, no. 1, pp. 2–5, Mar. 2019.
[104] H. H. Esmat and B. Lorenzo, ''Deep reinforcement learning based dynamic edge/fog network slicing,'' in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2020, pp. 1–6.
[105] V. Sciancalepore, L. Zanzi, X. Costa-Perez, and A. Capone, ''ONETS: Online network slice broker from theory to practice,'' 2018, *arXiv:1801.03484*.

[106] M. R. Raza, C. Natalino, L. Wosinska, and P. Monti, "Machine learning methods for slice admission in 5G networks," in *Proc. 24th OptoElectronics Commun. Conf. (OECC) Int. Conf. Photon. Switching Comput. (PSC)*, Jul. 2019, pp. 2019–2021.

[107] M. R. Raza, C. Natalino, P. Öhlen, L. Wosinska, and P. Monti, "A slice admission policy based on reinforcement learning for a 5G flexible RAN," in *Proc. Eur. Conf. Opt. Commun. (ECOC)*, no. 1, Sep. 2018, pp. 2–4.

[108] G. Dandachi, A. De Domenico, D. T. Hoang, and D. Niyato, "An artificial intelligence framework for slice deployment and orchestration in 5G networks," *IEEE Trans. Cogn. Commun. Netw.*, vol. 6, no. 2, pp. 858–871, Jun. 2020.

[109] T-Specification, "3rd generation partnership project; technical specification group services and system aspects; telecommunication management; study on management and orchestration of network slicing for next generation network (3GPP TS 28.801 version 15.1.0 Release 15)," Tech. Rep. TS 28.801, V15.0.0, 3rd Generation Partnership Project, Sophia Antipolis, France, 2018.

[110] F. Rezazadeh, H. Chergui, and C. Verikoukis, "Zero-touch continuous network slicing control via scalable actor-critic learning," 2021, *arXiv:2101.06654*.

[111] F. Rezazadeh, H. Chergui, L. Alonso, and C. Verikoukis, "Continuous multi-objective zero-touch network slicing via twin delayed DDPG and OpenAI gym," 2021, *arXiv:2101.06617*.

[112] F. Rezazadeh, H. Chergui, L. Christofi, and C. Verikoukis, "Actor-critic-based learning for zero-touch joint resource and energy control in network slicing," in *Proc. IEEE Int. Conf. Commun.*, Jun. 2021, pp. 1–6.

[113] M. Sulaiman, A. Moayyedi, M. A. Salahuddin, R. Boutaba, and A. Saleh, "Multi-agent deep reinforcement learning for slicing and admission control in 5G C-RAN," in *Proc. IEEE/IFIP Netw. Oper. Manag. Symp.*, Apr. 2022, pp. 1–9.

[114] W. Guan, H. Zhang, and V. C. M. Leung, "Customized slicing for 6G: Enforcing artificial intelligence on resource management," *IEEE Netw.*, vol. 35, no. 5, pp. 264–271, Sep. 2021.

[115] N. Van Huynh, D. T. Hoang, D. N. Nguyen, and E. Dutkiewicz, "Optimal and fast real-time resource slicing with deep dueling neural networks," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 6, pp. 1455–1470, Jun. 2019.

[116] T. Zhang, Y. Bian, Q. Lu, J. Qi, K. Zhang, H. Ji, W. Wang, and W. Wu, "Supervised learning based resource allocation with network slicing," in *Proc. 8th Int. Conf. Adv. Cloud Big Data, (CBD)*, 2020, pp. 25–30.

[117] A. Thantharate, R. Paropkari, V. Walunj, and C. Beard, "DeepSlice: A deep learning approach towards an efficient and reliable network slicing in 5G networks," in *Proc. IEEE 10th Annu. Ubiquitous Comput., Electron. Mobile Commun. Conf. (UEMCON)*, Feb. 2020, pp. 0762–0767.

[118] R. Xi, X. Chen, Y. Chen, and Z. Li, "Real-time resource slicing for 5G RAN via deep reinforcement learning," in *Proc. Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2019, pp. 625–632.

[119] H. Xiang, S. Yan, and M. Peng, "A realization of fog-RAN slicing via deep reinforcement learning," *IEEE Trans. Wireless Commun.*, vol. 19, no. 4, pp. 2515–2527, Apr. 2020.

[120] H. Xiang, M. Peng, Y. Sun, and S. Yan, "Mode selection and resource allocation in sliced fog radio access networks: A reinforcement learning approach," *IEEE Trans. Veh. Technol.*, vol. 69, no. 4, pp. 4271–4284, Apr. 2020.

[121] S. II Moon, H. Hirayama, Y. Tsukamoto, S. Nanba, and H. Shinbo, "Ensemble learning method-based slice admission control for adaptive RAN," in *Proc. IEEE Globecom Workshops (GC Wkshps*, Dec. 2020, pp. 1–6.

[122] B. Khodapanah, A. Awada, I. Viering, A. N. Barreto, M. Simsek, and G. Fettweis, "Slice management in radio access network via deep reinforcement learning," in *Proc. IEEE Veh. Technol. Conf.*, May 202, pp. 1–6.

[123] B. Khodapanah, A. Awada, I. Viering, A. N. Barreto, M. Simsek, and G. Fettweis, "Framework for slice-aware radio resource management utilizing artificial neural networks," *IEEE Access*, vol. 8, pp. 174972–174987, 2020.

[124] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, and X. Costa-Pérez, "A machine learning approach to 5G infrastructure market optimization," *IEEE Trans. Mobile Comput.*, vol. 19, no. 3, pp. 498–512, Mar. 2020.

[125] J. Mei, X. Wang, K. Zheng, G. Boudreau, A. B. Sediq, and H. Abou-Zeid, "Intelligent radio access network slicing for service provisioning in 6G: A hierarchical deep reinforcement learning approach," *IEEE Trans. Commun.*, vol. 69, no. 9, p. 6063–6078, Sep. 2021.

[126] B. Rebekka and B. Malarkodi, "Performance evaluation of resource allocation schemes in LTE downlink," in *Proc. Int. Conf. Electron. Commun. Syst. (ICECS)*, Feb. 2014, pp. 1–4.

[127] S. Matoussi, I. Fajjari, N. Aitsaadi, and R. Langar, "Deep learning based user slice allocation in 5G radio access networks," in *Proc. IEEE 45th Conf. Local Comput. Netw. (LCN)*, Jan. 2020, pp. 286–296.

[128] H. Zhang and V. W. S. Wong, "A two-timescale approach for network slicing in C-RAN," *IEEE Trans. Veh. Technol.*, vol. 69, no. 6, pp. 6656–6669, Jun. 2020.

[129] C. Qi, Y. Hua, R. Li, Z. Zhao, and H. Zhang, "Deep reinforcement learning with discrete normalized advantage functions for resource management in network slicing," *IEEE Commun. Lett.*, vol. 23, no. 8, pp. 1337–1341, Aug. 2019.

[130] D. Shome and A. Kudeshia, "Deep Q-learning for 5G network slicing with diverse resource stipulations and dynamic data traffic," in *Proc. Int. Conf. Artif. Intell. Inf. Commun. (ICAIIC)*, Apr. 2021, pp. 134–139.

[131] R. Li, Z. Zhao, Q. Sun, I. Chi-Lin, C. Yang, X. Chen, M. Zhao, and H. Zhang, "GAN-based deep distributional reinforcement learning for resource management in network slicing," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.

[132] S. A. Israel, J. Goldstein, J. S. Klein, J. Talamonti, F. Tanner, S. Zabel, P. A. Sallee, and L. McCoy, "Generative adversarial networks for classification," in *Proc. IEEE Appl. Imag. Pattern Recognit. Workshop (AIPR)*, Oct. 2017, pp. 1–4.

[133] R. Li, Z. Zhao, Q. Sun, C. Yang, X. Chen, M. Zhao, and H. Zhang, "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74429–74441, 2018.

[134] G. Chen, X. Mu, F. Shen, and Q. Zeng, "Network slicing resource allocation based on LSTM-D3QN with dual connectivity in heterogeneous cellular networks," *Appl. Sci.*, vol. 12, no. 18, p. 9315, 2022.

[135] S. Meng, Z. Wang, H. Ding, S. Wu, X. Li, P. Zhao, C. Zhu, and X. Wang, "RAN slice strategy based on deep reinforcement learning for smart grid," in *Proc. Comput., Commun. IoT Appl. (ComComAp)*, 2019, pp. 106–111.

[136] C. Zhang, M. Dong, and K. Ota, "Vehicular multi-slice optimization in 5G: Dynamic preference policy using reinforcement learning," in *Proc. IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–6.

[137] A. Nassar and Y. Yilmaz, "Deep reinforcement learning for adaptive network slicing in 5G for intelligent vehicular systems and smart cities," *IEEE Internet Things J.*, vol. 9, no. 1, pp. 222–235, Jan. 2022.

[138] J. Mei, X. Wang, and K. Zheng, "Intelligent network slicing for V2X services towards 5G," 2019, *arXiv:1910.01516*.

[139] N. Salhab, R. Rahim, R. Langar, and R. Boutaba, "Machine learning based resource orchestration for 5G network slices," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2019, pp. 1–6.

[140] L. Zanzi, V. Sciancalepore, A. Garcia-Saavedra, H. D. Schotten, and X. Costa-Perez, "LACO: A latency-driven network slicing orchestration in beyond-5G networks," *IEEE Trans. Wireless Commun.*, vol. 20, no. 1, pp. 667–682, Jan. 2021.

[141] Y. Shi, Y. E. Sagduyu, and T. Erpek, "Reinforcement learning for dynamic resource optimization in 5G radio access network slicing," in *Proc. IEEE 25th Int. Workshop Comput. Aided Model. Design Commun. Links Netw. (CAMAD)*, Sep. 2020, pp. 1–6.

[142] H. D. Albonda and J. Perez-Romero, "An efficient RAN slicing strategy for a heterogeneous network with eMBB and V2X services," *IEEE Access*, vol. 7, pp. 44771–44782, 2019.

[143] H. Khan, M. M. Butt, S. Samarakoon, P. Sehier, and M. Bennis, "Deep learning assisted CSI estimation for joint URLLC and eMBB resource allocation," 2020, *arXiv:2003.05685*.

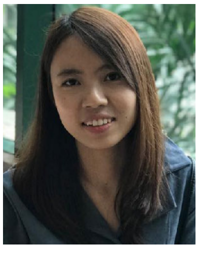
[144] J. Tang, Y. Duan, Y. Zhou, and J. Jin, "Distributed slice selection-based computation offloading for intelligent vehicular networks," *IEEE Open J. Veh. Technol.*, vol. 2, pp. 261–271, 2021.

[145] M. Zambianco and G. Verticale, "Spectrum allocation for network slices with inter-numerology interference using deep reinforcement learning," in *Proc. IEEE 31st Annu. Int. Symp. Pers., Indoor Mobile Radio Commun.*, Aug. 2020, pp. 1–7.

[146] Y. Shao, R. Li, B. Hu, Y. Wu, Z. Zhao, and H. Zhang, "Graph attention network-based multi-agent reinforcement learning for slicing resource management in dense cellular network," *IEEE Trans. Veh. Technol.*, vol. 70, no. 10, pp. 10792–10803, Oct. 2021.

[147] Y. Abiko, T. Saito, D. Ikeda, K. Ohta, T. Mizuno, and H. Mineno, ''Flexible resource block allocation to multiple slices for radio access network slicing using deep reinforcement learning,'' *IEEE Access*, vol. 8, pp. 68183–68198, 2020.
[148] J. A. Ayala-Romero, A. Garcia-Saavedra, M. Gramaglia, X. Costa-Perez, A. Banchs, and J. J. Alcaraz, ''vrAIn: Deep learning based orchestration for computing and radio resources in vRANs,'' *IEEE Trans. Mobile Comput.*, vol. 1233, no. 7, pp. 2652–2670, Jul. 2020.
[149] H. Chergui and C. Verikoukis, ''OPEX-limited 5G RAN slicing: An over-dataset constrained deep learning approach,'' in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jun. 2020, pp. 4–9.
[150] Y. Xu, Z. Zhao, P. Cheng, Z. Chen, M. Ding, B. Vucetic, and Y. Li, ''Constrained reinforcement learning for resource allocation in network slicing,'' *IEEE Commun. Lett.*, vol. 25, no. 5, pp. 1554–1558, May 2021.
[151] Y. Liu, J. Ding, and X. Liu, ''A constrained reinforcement learning based approach for network slicing,'' in *Proc. IEEE 28th Int. Conf. Netw. Protocols (ICNP)*, Oct. 2020, pp. 1–6.
[152] X. Chen, Z. Zhao, C. Wu, M. Bennis, H. Liu, Y. Ji, and H. Zhang, ''Multi-tenant cross-slice resource orchestration: A deep reinforcement learning approach,'' *IEEE J. Sel. Areas Commun.*, vol. 37, no. 10, pp. 2377–2392, Oct. 2019.
[153] S. Messaoud, A. Bradai, O. B. Ahmed, P. T. A. Quang, M. Atri, and M. S. Hossain, ''Deep federated Q-learning-based network slicing for industrial IoT,'' *IEEE Trans. Ind. Informat.*, vol. 17, no. 8, pp. 5572–5582, Aug. 2021.
[154] A. Gharsallah, N. Omheni, K. Ghanmi, F. Zarai, and M. Neji, ''A seamless mobility mechanism for V2V communications,'' in *Proc. IEEE/ACS 14th Int. Conf. Comput. Syst. Appl. (AICCSA)*, Oct. 2017, pp. 1063–1069.
[155] R. Molina-Masegosa and J. Gozalvez, ''LTE-V for sidelink 5G V2X vehicular communications: A new 5G technology for short-range vehicle-to-everything communications,'' *IEEE Veh. Technol. Mag.*, vol. 12, no. 4, pp. 30–39, Dec. 2017.
[156] A. B. Kihero, M. S. J. Solaija, A. Yazar, and H. Arslan, ''Inter-numerology interference analysis for 5G and beyond,'' in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Feb. 2018, pp. 1–6.
[157] S. D. Oro, L. Bonati, S. Member, F. Restuccia, T. Melodia, and S. Member, ''Coordinated 5G network slicing: How constructive interference can boost network throughput,'' *IEEE/ACM Trans. Netw.*, vol. 29, no. 4, pp. 1881–1894, Aug. 2021.
[158] D. Horgan, J. Quan, D. Budden, G. Barth-Maron, M. Hessel, H. van Hasselt, and D. Silver, ''Distributed prioritized experience replay,'' in *Proc. ICLR*, 2018, pp. 1–19.
[159] N. Yu, Z. Song, H. Du, H. Huang, and X. Jia, ''Multi-resource allocation in cloud radio access networks,'' in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jul. 2017, pp. 1–6.
[160] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, ''Cloud RAN for mobile networks—A technology overview,'' *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 405–426, 1st Quart., 2015.
[161] G. Sun, K. Xiong, G. O. Boateng, G. Liu, and W. Jiang, ''Resource slicing and customization in RAN with dueling deep Q-network,'' *J. Netw. Comput. Appl.*, vol. 157, May 2020, Art. no. 102573, doi: 10.1016/j.jnca.2020.102573.
[162] K. Yu, H. Zhou, B. Qian, Z. Tang, and X. Shen, ''A reinforcement learning aided decoupled RAN slicing framework for cellular V2X,'' in *Proc. IEEE Global Commun. Conf.*, Jan. 2020, pp. 2–7.
[163] Q. Liu, T. Han, N. Zhang, and Y. Wang, ''DeepSlicing: Deep reinforcement learning assisted resource allocation for network slicing,'' 2020, *arXiv:2008.07614*.
[164] G. Sun, G. O. Boateng, D. Ayepah-Mensah, G. Liu, and J. Wei, ''Autonomous resource slicing for virtualized vehicular networks with D2D communications based on deep reinforcement learning,'' *IEEE Syst. J.*, vol. 14, no. 4, pp. 4694–4705, Dec. 2020.
[165] Y. Cui, X. Huang, P. He, D. Wu, and R. Wang, ''A two-timescale resource allocation scheme in vehicular network slicing,'' in *Proc. IEEE 93rd Veh. Technol. Conf. (VTC-Spring)*, Jun. 2021, pp. 1–5.
[166] F. Wei, G. Feng, Y. Sun, Y. Wang, S. Qin, and Y. C. Liang, ''Network slice reconfiguration by exploiting deep reinforcement learning with large action space,'' *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 4, pp. 2197–2211, Dec. 2020.
[167] X. Zhang, W. Lu, B. Li, and Z. Zhu, ''DRL-based network orchestration to realize cooperative, distributed and tenant-driven virtual network slicing,'' in *Proc. Asia Commun. Photon. Conf. (ACP)*, 2019, pp. 2019–2021.
[168] W. Lu, H. Fang, and Z. Zhu, ''AI-assisted resource advertising and pricing to realize distributed tenant-driven virtual network slicing in inter-DC optical networks,'' in *Proc. 22nd Conf. Opt. Netw. Design Model. (ONDM)*, 2018, pp. 130–135.
[169] T. Li, X. Zhu, and X. Liu, ''An end-to-end network slicing algorithm based on deep Q-learning for 5G network,'' *IEEE Access*, vol. 8, pp. 122229–122240, 2020.
[170] G. Yang, Q. Liu, X. Zhou, Y. Qian, and W. Wu, ''Two-tier resource allocation in dynamic network slicing paradigm with deep reinforcement learning,'' in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Feb. 2019, pp. 1–6.
[171] F. Yang, W. Wu, X. Wang, Y. Zhang, and P. Si, ''Deep reinforcement learning based handoff algorithm in end-to-end network slicing enabling HetNets,'' in *Proc. IEEE Wireless Commun. Netw. Conf. (WCNC)*, Mar. 2021, pp. 1–7.
[172] Y. Sun, W. Jiang, G. Feng, P. V. Klaine, L. Zhang, M. A. Imran, and Y.-C. Liang, ''Efficient handover mechanism for radio access network slicing by exploiting distributed learning,'' *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 4, pp. 2620–2633, Dec. 2020.
[173] A. Laghrissi and T. Taleb, ''A survey on the placement of virtual resources and virtual network functions,'' *IEEE Commun. Surveys Tuts.*, vol. 21, no. 2, pp. 1409–1434, 2nd Quart., 2019.
[174] G. Wang, G. Feng, T. Q. Quek, S. Qin, R. Wen, and W. Tan, ''Reconfiguration in network slicing—Optimizing the profit and performance,'' *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 2, pp. 591–605, Jun. 2019.
[175] M. Pozza, P. K. Nicholson, D. F. Lugones, A. Rao, H. Flinck, and S. Tarkoma, ''On reconfiguring 5G network slices,'' *IEEE J. Sel. Areas Commun.*, vol. 38, no. 7, pp. 1542–1554, Jul. 2020.
[176] A. Tavakoli, F. Pardo, and P. Kormushev, ''Action branching architectures for deep reinforcement learning,'' 2017, *arXiv:1711.08946*.
[177] C. Kachris and I. Tomkos, ''Optical interconnection networks for data centers,'' in *Proc. 17th Int. Conf. Opt. Netw. Design Model. (ONDM)*, 2013, pp. 19–22.
[178] X. Xue, F. Wang, F. Agraz, A. Pagès, B. Pan, F. Yan, X. Guo, S. Spadaro, and N. Calabretta, ''SDN-controlled and orchestrated OPSquare DCN enabling automatic network slicing with differentiated QoS provisioning,'' *J. Lightw. Technol.*, vol. 38, no. 6, pp. 1103–1112, Mar. 2020.
[179] H. Jiang, Y. Wang, L. Gong, and Z. Zhu, ''Availability-aware survivable virtual network embedding in optical datacenter networks,'' *J. Opt. Commun. Netw.*, vol. 7, no. 12, pp. 1160–1171, Dec. 2015.
[180] Y.-J. Liu, G. Feng, J. Wang, Y. Sun, and S. Qin, ''Access control for ran slicing based on federated deep reinforcement learning,'' in *Proc. IEEE Int. Conf. Commun.*, Aug. 2021, pp. 1–6.
[181] C. Mei, J. Liu, J. Li, L. Zhang, and M. Shao, ''5G network slices embedding with sharable virtual network functions,'' *J. Commun. Netw.*, vol. 22, no. 5, pp. 415–427, 2020.
[182] D. Harutyunyan, R. Fedrizzi, N. Shahriar, R. Boutaba, and R. Riggio, ''Orchestrating end-to-end slices in 5G networks,'' in *Proc. 15th Int. Conf. Netw. Service Manage. (CNSM)*, Oct. 2019, pp. 1–9.
[183] H. Wang, Y. Wu, G. Min, J. Xu, and P. Tang, ''Data-driven dynamic resource scheduling for network slicing: A Deep reinforcement learning approach,'' *Inf. Sci.*, vol. 498, pp. 106–116, Sep. 2019, doi: 10.1016/j.ins.2019.05.012.
[184] F. Fossati, S. Moretti, P. Perny, and S. Secci, ''Multi-resource allocation for network slicing,'' *IEEE/ACM Trans. Netw.*, vol. 28, no. 3, pp. 1311–1324, Jun. 2020.
[185] M. Iannelli, M. R. Rahman, N. Choi and L. Wang, ''Applying machine learning to end-to-end slice SLA decomposition,'' in *Proc. 6th IEEE Conf. Netw. Softwarization (NetSoft)*, Jul. 2020, pp. 92–99.
[186] G. Kibalya, J. Serrat, J. L. Gorricho, R. Pasquini, H. Yao, and P. Zhang, ''A reinforcement learning based approach for 5G network slicing across multiple domains,'' in *Proc. 15th Int. Conf. Netw. Service Manage. (CNSM)*, 2019, pp. 1–5.
[187] W. Guan, H. Zhang, and V. C. Leung, ''Slice reconfiguration based on demand prediction with dueling deep reinforcement learning,'' in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Jan. 2020, pp. 1–6.
[188] A. Rkhami, Y. Hadjadj-Aoul, and A. Outtagarts, ''Learn to improve: A novel deep reinforcement learning approach for beyond 5G network slicing,'' in *Proc. IEEE 18th Annu. Consum. Commun. Netw. Conf. (CCNC)*, Jul. 2021, pp. 1–6.
[189] J. J. A. Esteves, A. Boubendir, F. Guillemin, and P. Sens, ''On the robustness of controlled deep reinforcement learning for slice placement,'' *J. Netw. Syst. Manage.*, vol. 30, no. 3, pp. 1–10, 2022.

[190] R. A. Addad, D. L. C. Dutra, T. Taleb, and H. Flinck, "AI-based network-aware service function chain migration in 5G and beyond networks," *IEEE Trans. Netw. Service Manage.*, vol. 19, no. 1, pp. 472–484, Mar. 2022.
[191] F. Zhou, P. Yu, L. Feng, X. Qiu, Z. Wang, L. Meng, M. Kadoch, L. Gong, and X. Yao, "Automatic network slicing for IoT in smart city," *IEEE Wireless Commun.*, vol. 27, no. 6, pp. 108–115, Dec. 2020.
[192] A. Ksentini, M. Jebalia, and S. Tabbane, "Fog-enabled industrial IoT network slicing model based on ML-enabled multi-objective optimization," in *Proc. IEEE 29th Int. Conf. Enabling Technol., Infrastructure Collaborative Enterprises (WETICE)*, Sep. 2021, pp. 177–180.
[193] C.-C. Wu, V. Friderikos, and C. Stefanovic, "Multi-objective provisioning of network slices using deep reinforcement learning," 2022, *arXiv:2207.13821*.
[194] N. Van Huynh, D. T. Hoang, D. N. Nguyen, and E. Dutkiewicz, "Real-time network slicing with uncertain demand: A deep learning approach," in *Proc. IEEE Int. Conf. Commun.*, May 2019, pp. 1–6.
[195] J. Koo, V. B. Mendiratta, M. R. Rahman, and A. Walid, "Deep reinforcement learning for network slicing with heterogeneous resource requirements and time varying traffic dynamics," in *Proc. 15th Int. Conf. Netw. Service Manage. (CNSM)*, 2019, pp. 1–5.
[196] F. Mason, G. Nencioni, and A. Zanella, "A multi-agent reinforcement learning architecture for network slicing orchestration," in *Proc. 19th Medit. Commun. Comput. Netw. Conf. (MedComNet)*, 2021, pp. 1–8.
[197] Q. Liu, T. Han, and E. Moges, "EdgeSlice: Slicing wireless edge computing network with decentralized deep reinforcement learning," 2020, *arXiv:2003.12911*.
[198] S. A. Rahman, H. Tout, H. Ould-Slimane, A. Mourad, C. Talhi, and M. Guizani, "A survey on federated learning: The journey from centralized to distributed on-site learning and beyond," *IEEE Internet Things J.*, vol. 8, no. 7, pp. 5476–5497, Apr. 2020.
[199] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Long Beach, CA, USA. Red Hook, NY, USA: Curran, 2017, pp. 6000–6010.
[200] N. Wu, B. Green, X. Ben, and S. O'Banion, "Deep transformer models for time series forecasting: The influenza prevalence case," 2020, *arXiv:2001.08317*.
[201] T. T. Nguyen, N. D. Nguyen, and S. Nahavandi, "Deep reinforcement learning for multiagent systems: A review of challenges, solutions, and applications," *IEEE Trans. Cybern.*, vol. 50, no. 9, pp. 3826–3839, Sep. 2020.
[202] C. Liu, X. Xu, and D. Hu, "Multiobjective reinforcement learning: A comprehensive overview," *IEEE Trans. Syst. Man, Cybern, Syst.*, vol. 45, no. 3, pp. 385–398, Mar. 2015.
[203] S. Lange and M. Riedmiller, "Deep auto-encoder neural networks in reinforcement learning," in *Proc. 2010 Int. Joint Conf. Neural Netw. (IJCNN)*, 2010, pp. 1–8.
[204] N. Kornfeld and Z. Feng, "A latent variable model state estimation system for image sequences," in *Proc. 22th Int. Conf. Inf. Fusion (FUSION)*, 2019, pp. 1–8.
[205] A. Hussein, M. M. Gaber, E. Elyan, and C. Jayne, "Imitation learning: A survey of learning methods," *ACM Comput. Surveys*, vol. 50, no. 2, pp. 1–35, Apr. 2017, doi: 10.1145/3054912.
[206] T. M. Hospedales, A. Antoniou, P. Micaelli, and A. J. Storkey, "Meta-learning in neural networks: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 1, pp. 5149–5169, May 5555.
[207] S. Saha, Z. Gan, L. Cheng, J. Gao, O. L. Kafka, X. Xie, H. Li, M. Tajdari, H. A. Kim, and W. K. Liu, "Hierarchical deep learning neural network (HiDeNN): An artificial intelligence (AI) framework for computational science and engineering," *Comput. Methods Appl. Mech. Eng.*, vol. 373, Oct. 2021, Art. no. 113452.
[208] K. Kowsari, D. E. Brown, M. Heidarysafa, K. Jafari Meimandi, M. S. Gerber, and L. E. Barnes, "HDLTex: Hierarchical deep learning for text classification," in *Proc. 16th IEEE Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2017, pp. 1–7, doi: 10.1109/ICMLA.2017.0-134.
[209] L. Zhang, L. Cheng, H. Li, J. Gao, C. Yu, R. Domel, Y. Yang, S. Tang, and W. K. Liu, "Hierarchical deep-learning neural networks: Finite elements and beyond," *Comput. Mech.*, vol. 67, no. 1, pp. 207–230, Jan. 2021, doi: 10.1007/s00466-020-01928-9.
[210] H. Chergui, L. Blanco, L. A. Garrido, and K. Ramantas, "Zero-touch AI-driven distributed management for energy-efficient 6G massive network slicing," *IEEE Netw.*, vol. 35, no. 6, pp. 43–49, Nov./Dec. 2021.
[211] M. A. Hossain, A. R. Hossain, and N. Ansari, "AI in 6G: Energy-efficient distributed machine learning for multilayer heterogenous networks," *IEEE Netw.*, vol. 36, no. 6, pp. 84–91, Dec. 2022.
[212] H. Larsson, H. Riaz, and S. Ickin, "Automated collaborator selection for federated learning with multi-armed bandit agents," *Proc. 4th FlexNets Workshop Flexible Networks, Artif. Intell. Supported Netw. Flexibility Agility (SIGCOMM)*, 2021, pp. 44–49.



HNIN PANN PHYU (Student Member, IEEE) received the M.Sc. degree in communication networks and services from the Institut Mines-Telecom, France, in 2016, and the M.Eng. degree in telecommunications from the Asian Institute of Technology, Thailand, in 2016. She is currently pursuing the Ph.D. degree in software and IT engineering with École de technologie supérieure University, Montreal, Canada. From 2016 to 2020, she was a Network Strategist and an Architect Engineer with Telenor, Myanmar. Her current research interests include next generation mobile communication systems, machine learning, big data analytics, resource management, and network slicing.

DIALA NABOULSI (Member, IEEE) received the M.Sc. degree in computer science from INSA Lyon, France, the M.Eng. degree in telecommunications from Lebanese University, Lebanon, in 2012, and the Ph.D. degree in computer science from INSA Lyon, in 2015. She was a Research Professional in the Ultra-TCS Research Chair with École de Technologie Supérieure (ÉTS), Canada, and a Research Associate and a Postdoctoral Researcher with Concordia University, Canada. She was also a Visiting Researcher with Ericsson, Canada. She was a Course Lecturer with McGill University, Canada, and Concordia University. She is currently an Assistant Professor with ÉTS. Her research interests include mobile networks, virtualized networks, and wireless networks. She was the Holder of an NSERC Discovery Grant (2020–2026), on network slicing in future mobile networks. She has collaborations with several providers in communications technology, such as ultra intelligence and communications and Octasic.

RAZVAN STANICA received the M.Eng. and Ph.D. degrees in computer science from INP Toulouse, France, in 2008 and 2011, respectively. He is currently an Associate Professor with INSA Lyon, France, and a Research Scientist with the CITI Laboratory, Inria Agora Team. His research interest includes wireless mobile networks, with a special focus on communication networks in urban environments.