Survey on Machine Learning-Enabled Network Slicing: Covering the Entire Life Cycle

Adnei Donatti, Sand L. Corrêa, Joberto S. B. Martins, *Life Senior Member, IEEE*, Antonio J. G. Abelem, *Member, IEEE*, Cristiano Bonato Both, Flávio de Oliveira Silva, *Member, IEEE*, José A. Suruagy, *Member, IEEE*, Rafael Pasquini, *Senior Member, IEEE*, Rodrigo Moreira, Kleber V. Cardoso, and Tereza C. Carvalho

***Abstract*—Network slicing (NS) is becoming an essential element of service management and orchestration in communication networks, starting from mobile cellular networks and extending to a global initiative. NS can reshape the deployment and operation of traditional services, support the introduction of new ones, vastly advance how resource allocation performs in networks, and notably change the user experience. Most of these promises still need to reach the real world, but they have already demonstrated their capabilities in many experimental infrastructures. However, complexity, scale, and dynamism are pressuring for a Machine Learning (ML)-enabled NS approach in which autonomy and efficiency are critical features. This trend is relatively new but growing fast and attracting much attention. This article surveys Artificial Intelligence-enabled NS and its potential use in current and future infrastructures. We have covered state-of-the-art ML-enabled NS for all network segments and organized the literature according to the phases of the NS life cycle. We also discuss challenges and opportunities in research on this topic.**

***Index Terms*—Network slicing, ML-enabled slicing, machine learning, slicing-as-a-service, ML-enabled resource orchestration, and allocation.**

Manuscript received 15 November 2022; revised 3 May 2023; accepted 14 June 2023. Date of publication 20 June 2023; date of current version 7 February 2024. The authors thank the National Council for Scientific and Technological Development (CNPQ), ANIMA Institute, CNPq Grant 421944/2021-8 and FAPESP MCTIC/CGI cooperation agreement under the thematic research project 2018/23097-3 - SFI2 - Slicing Future Internet Infrastructures. The associate editor coordinating the review of this article and approving it for publication was B. Martini. *(Corresponding author: Joberto S. B. Martins.)*

Adnei Donatti and Tereza C. Carvalho are with the Department of Computer Engineering and Digital Systems, University of São Paulo, São Paulo 05508-020, Brazil (e-mail: adnei.donatti@usp.br; terezacarvalho@usp.br).

Sand L. Corrêa and Kleber V. Cardoso are with the Institute of Informatics, Federal University of Goiás, Goiânia 74001-970, Brazil (e-mail: sand@inf.ufg.br; kleber@inf.ufg.br).

Joberto S. B. Martins is with the Computer Science Department, Salvador University, Salvador 41830-500, Brazil (e-mail: joberto@ieee.org).

Antonio J. G. Abelem is with the Computer Science Department, Federal University of Pará, Belém 66075-110, Brazil (e-mail: abelem@ufpa.br).

Cristiano Bonato Both is with PPGCA, UNISINOS, São Leopoldo 93022-000, Brazil (e-mail: cbboth@unisinos.br).

Flávio de Oliveira Silva and Rafael Pasquini are with the Faculty of Computing, Federal University of Uberlândia, Uberlândia 38400902, Brazil (e-mail: flavio@ufu.br; rafael.pasquini@gmail.com).

José A. Suruagy is with the Informatics Center, Federal University of Pernambuco, Recife 52041-800, Brazil (e-mail: suruagy@cin.ufpe.br).

Rodrigo Moreira is with the Institute of Exact and Technological Sciences, Federal University of Viçosa, Viçosa 36570-000, Brazil (e-mail: rodrigo@ufv.br).

Digital Object Identifier 10.1109/TNSM.2023.3287651

I. INTRODUCTION

OVER the last decade, wireless networking technology has been mainly driven by advanced networking applications such as Industry 4.0, immersive media applications (e.g., virtual/augmented/mixed reality), and mission-critical services (e.g., self-driving vehicles and automated traffic control systems) [1]. Following this trend, the fifth-generation (5G) cellular networks have been designed to provide higher latency, bit rate, and reliability performance, fostering the digital transformation of vertical industries [2]. A requirement to achieve this goal is to support different communication services, e.g., Machine Type Communication (mMTC), Enhanced Mobile Broadband (eMBB), ultra-Reliable and Low-Latency Communications (URLLC), with highly different needs, over a shared network infrastructure [3]. To address this challenge, 5G and beyond 5G networks embrace the concept of Network Slicing (NS) [4], [5], [6], which logically divides the operator's network into isolated, service-tailored, end-to-end networks referred to as *network slices*. The NS concept brings several advantages to network operators [7]. First, NS allows multiple tenants to share the same physical network infrastructure and reduce network deployment and operation costs. Second, with NS, each network slice is instantiated to satisfy a particular set of applications, enabling service differentiation and guaranteeing Service Level Agreement (SLA) for each application type. Finally, NS increases flexibility in network management, as network slices can be created, modified, and decommissioned as needed. However, to fully exploit the advantages of NS, operators have to provide dynamic resource allocation, service assurance, isolation and protection, and optimized partitioning of resources across all network domains, i.e., Radio Access Network (RAN), Transport Network (TN), and Core Network (CN), and throughout the entire slice life cycle, from the slice preparation to the slice decommissioning. Therefore, the benefits of NS come at the price of higher complexity in operating and managing wireless networks.

Currently, the realization of the NS concept relies heavily on paradigms such as Network Function Virtualization (NFV), Software-Defined Network (SDN), and cloud computing. Together, these technologies provide the means of control for dynamically allocating the necessary resource capacities across the network and resizing and moving workloads at runtime to meet the needs of services, regardless of network conditions [14]. However, although these means of control are already available, the decision-making process that triggers their execution depends on static policies and human intervention [1], [22]. Therefore, the full realization of the NS paradigm depends on further automation and the closure of management control loops.

Due to advances in algorithms and the increase in computational power, in recent years, Artificial Intelligence (AI), and Machine Learning (ML) in particular, has become an essential enabling technology to achieve good performance in complex decision-making problems [23]. Indeed, ML techniques are enablers of numerous problems involving multiple objectives subject to many heterogeneous and dynamic requirements [7], [24]. NS, in turn, is a current trend that, as a problem, inherently has multiple objectives, potentially deals with many domains and technologies, and supports numerous users and heterogeneous requirements. Therefore several works have applied ML to deal with distinct challenges during the slice life cycle. Yang et al. [25] proposed an intent-driven optical NS that maps high-level intents into slice requirements for the transport network using Latent Dirichlet Allocation. Sciancalepore et al. [26] designed an online network slice broker that decides which slices to accept while opportunistically pursuing the NS multiplexing gain maximization using a variant of the Multi-Armed Bandit (MAB) model. Kibalya et al. [27] formulated the multi-domain slicing as a multisubstrate Virtual Network Embedding (VNE) problem and proposed a Deep Reinforcement Learning (DRL) algorithm to solve it. Bega et al. [28] proposed a Deep Learning (DL) algorithm that anticipates future slice needs and timely reallocates/deallocates resources where and when they are required. Although these works have shown the potential of ML for supporting the emerging need for autonomous network slice operation and management, the literature has only unsystematically addressed individual problems. Consequently, there is a need to investigate and reorganize the current proposals for a comprehensive view of the fundamental network slice Life Cycle Management (LCM) problems and the existing ML proposals to deal with them.

A. Related Surveys

Several existing surveys have discussed the implications of the NS concept for next-generation mobile networks. Foukas et al. [6], Afolabi et al. [8], Kaloxylos [9], Zhang [10], and Khan et al. [11] provided the research community with a general understanding of the topic, addressing NS in terms of basic concepts, enabling technologies, use cases, and challenges.

Some surveys have discussed the implementation aspects of NS. Barakabitze et al. [12] provided a comprehensive review of solutions for NS using SDN and NFV. Various 5G architectural approaches were compared in terms of practical implementations in their work. Chahbar et al. [13] and Ordonez-Lucena et al. [14] focused on the ongoing work on NS modeling in RAN, TN, and CN domains performed by different Standards Developing Organization (SDO). Wijethilaka and Liyanage [15] studied the contribution of NS to the Internet of Things (IoT) realization.

The algorithmic aspects of NS have also been discussed in the literature [16], [17], [18]. Specifically, Vassilaras et al. [16] formulated NS as an optimization problem of placing Virtualized Network Functions (VNFs) over a set of candidate locations and deciding their interconnections. Su et al. [17] surveyed the resource allocation schemes for NS using three mathematical models: game theory, prediction techniques, and robustness/failure recovery models. Debbabi et al. [18] reviewed the state-of-the-art NS regarding two algorithmic challenges: slice resource allocation and slice orchestration. Nevertheless, these surveys considered only a few algorithmic aspects of NS, and none focused on ML solutions. Indeed, the need to use ML for network slice operation and management was first discussed by Kafle et al. [20]. The authors described the management functions of network slices that could be automated using ML and listed relevant techniques for automating such functions. However, the authors did not survey existing works and proposed solutions. More recently, Shen et al. [7] surveyed ML solutions applied to intelligent NS management. Nevertheless, the authors considered only three specific RAN problems: flexible radio access NS, automatic Radio Access Technology (RAT) selection, and mobile edge caching and content delivery. Wu et al. [21] discussed a broad picture of the role of AI in sixth-generation (6G) networks, highlighting potential NS problems where AI could be applied to facilitate intelligent network management. However, similar to [20], the authors did not survey existing works and proposed solutions. Ssengonzi et al. [19] presented a survey of 5G NS and virtualization from a Reinforcement Learning (RL) and DRL perspective. Nevertheless, the authors focused only on existing RL and DRL approaches and a few NS problems, such as resource allocation, admission control, and traffic forecasting.

Table I summarizes the main characteristics of existing surveys and our work, comparing them in terms of their main focus (i.e., NS concepts, NS implementation aspects, or NS algorithmic aspects), whether ML is considered, whether existing solutions are discussed, and the main criteria driving the study. As illustrated in the table, a comprehensive survey of ML applied to solve network slice LCM problems is still missing.

B. Research Scope and Methodology

The primary goal of this work is to provide the reader with a comprehensive survey of the use of ML for intelligent network slice LCM, from the slice preparation to their decommissioning, after the 3rd Generation Partnership Project (3GPP) life cycle [29] and covering all network domains (RAN, TN, and CN). We studied and assessed high-quality research published since 2016, available in the vehicles Institute of Electrical and Electronics Engineers (IEEE)Xplore, Association for Computing Machinery (ACM) Digital Library, Science Direct, and Wiley Online Library. We introduce the existing works in terms of the problem they address (e.g., slice admission control, resource allocation, VNF placement) after the 3GPP slice life cycle. Fig. 1 illustrates the organization

TABLE I
COMPARISON OF RELATED SURVEYS

Paper	Main focus	Focus on ML	Existing works are surveyed	Study's Orientation
[6]	NS concepts	✗	✓	Background-oriented
[8]	NS concepts	✗	✓	Background-oriented
[9]	NS concepts	✗	✓	Background-oriented
[10]	NS concepts	✗	✗	Background-oriented
[11]	NS concepts	✗	✓	Background-oriented
[12]	NS implementation aspects	✗	✓	SDN and NFV-oriented
[13]	NS implementation aspects	✗	✓	SDO-oriented
[14]	NS implementation aspects	✗	✓	Network segment-oriented
[15]	NS implementation aspects	✗	✓	IoT-oriented
[16]	NS algorithmic aspects	✗	✗	VNF placement-oriented
[17]	NS algorithmic aspects	✗	✓	Resource allocation perspective
[18]	NS algorithmic aspects	✗	✓	Resource allocation perspective
[19]	NS algorithmic aspects	✓	✓	Resource allocation perspective
[7]	NS algorithmic aspects	✓	✓	RAN-oriented
[20]	NS algorithmic aspects	✓	✗	LCM-oriented
[21]	NS algorithmic aspects	✓	✗	LCM-oriented
This survey	NS algorithmic aspects	✓	✓	LCM-oriented

of the article while Table V summarizes the commonly-used abbreviations.

At the beginning of the slice life cycle, i.e., in the preparation phase, ML is mainly employed to translate service profiles into slice requirements and to provide a slice admission control. In the commissioning phase, ML is applied for slice resource allocation, slice VNF placement, and slice path configuration. Next, when the slice becomes operational, ML is employed for numerous runtime tasks, including user admission control, task offloading, slice elasticity, anomaly detection, RAT selection, traffic classification and prediction, congestion control, and mobility management. We point out that our survey did not find works applying ML to problems related to slicing termination, i.e., to the decommissioning phase. Therefore, decommissioning is not illustrated in Fig. 1. In addition, our survey classifies each article according to the main problem it addresses, even when the article focuses on multiple problems and life cycle phases. For example, some research efforts describe using AI in two or more life cycle phases, such as [30]. We classify such works according to their main addressed problem. Finally, although some works have proposed solutions to network slice LCM problems using heuristic and genetic algorithms, our survey focuses on supervised, unsupervised, reinforcement, and emerging learning paradigms. In summary, our main contribution as a survey is to bring a big picture of the state-of-the-art ML-enabled NS and organize the existing works from the network slice life cycle perspective, illustrating the AI/ML methods used in the distinct phases of the network slice life cycle. Therefore, we carefully analyze every article and decide the life cycle phase it fits based on the problem addressed.

This article is organized as follows. Section II presents an overview of the main concepts related to NS, focusing on NS management. Next, we thoroughly review the state-of-the-art solutions for intelligent NS management. We split the related discussion into ML-enable solutions for NS problems during the preparation phase (Section III), the commissioning phase (Section IV), and the operation phase (Section V). We discuss

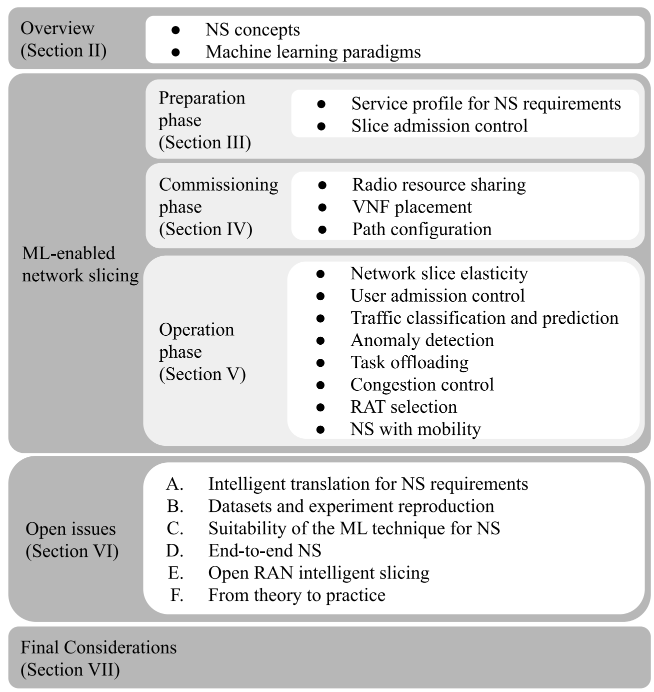

Fig. 1. The structure of the survey.

some open research issues and summarize potential future directions in Section VI. Finally, Section VII concludes the article.

II. OVERVIEW AND BACKGROUND

This section focuses on background concepts and key entities related to NS implementation and slice LCM to get deeper insights into the NS life cycle problems. Following the nomenclature proposed by 3GPP, a network slice, or slice, is a logical network comprising one or more service chains formed by virtualized or physical network functions and the (physical/virtual) links connecting them. This logical network is created with appropriate isolation, resources, and optimized topology to serve one or more communication services [29]. *Communication service* is the term used to refer to the tenant-ordered service. Usually, the communication service is expressed by a *service profile* comprising the service type, and a service graph, where nodes represent computing/storage resources and service instances and edges denote constraints on link bandwidth or packet loss. A network slice can host multiple communication services if they do not impose conflicting requirements.

A network slice is an end-to-end concept, i.e., the logical network can span across all the technical domains (or segments) within the operator's network, including the RAN, TN, and CN domains. In the 5G architecture, the RAN domain connects User Equipment (UE) to the operator's network using various access technologies. The TN domain provides infrastructure connectivity between the RAN and the data network using any technology (Internet Protocol (IP), optical, microwave, or other technology), tunnel (IP/Multiprotocol Label Switching (MPLS)), and layer functions [13]. Finally, the CN domain allows UE to send/receive data to/from the data network, providing signaling procedures such as connection, registration, mobility management, and session management. The part of the slice spanning a technical domain is called a Network Slice Subnet (NSS). Each NSS is usually deployed as a set of network functions.

Since a network slice can host multiple services, its life cycle within the operator's network is independent of its associated service(s) life cycle. In particular, the life cycle of a network slice has four main phases: *Preparation*, *Commissioning*, *Operation*, and *Decommissioning*. In the preparation phase, the slice does not exist. Indeed, the preparation phase comprises all planning steps that precede slice instantiation, such as slice design, slice onboarding, and slice admission control. In the first step, the tenant defines the service profile from which the *slice requirements* will be derived. In the second step, the tenant uploads the VNFs that constitute the slice to the operator's system. The last step in the preparation phase decides whether the tenant NS request should be accepted or rejected based on current system utilization. In the commissioning phase, resources are assigned to the admitted slice request. Therefore, the slice is instantiated, configured, and activated over the operator's infrastructure according to its requirements. In the operation phase, the slice instance goes into operation, and its behavior is monitored to ensure compliance with the defined requirements. In this phase, runtime tasks such as upgrade, reconfiguration, scaling, and capacity changes can be carried out to modify the slice instance and ensure that it is optimized for its purpose. Finally, the slice instance is terminated in the decommissioning phase, and its allocated resources are released.

Since our focus is on AI/ML solutions to NS problems, it is imperative to introduce the ML paradigms, which are traditionally classified into three types: Supervised Learning (SL), Unsupervised Learning (UL), and RL. SL uses labeled training datasets to build models and is usually employed to solve classification and regression problems to predict outcomes. UL creates models using unlabeled training datasets, mainly employed for clustering problems. In RL, an agent interacts with the environment via perception and action to learn a reward or utility. Therefore, an RL agent learns by exploring the environment instead of being taught by exemplars. The literature has applied the aforementioned paradigms to solve some of the NS problems we cover in this survey. In addition, emergent learning paradigms such as Federated Learning (FL) and Transfer Learning (TL) have also been employed in some works. FL focuses on decentralization learning, where distributed servers train models with local data. TL aims to utilize the built knowledge of a certain system to solve a different but related problem. We refer readers unfamiliar with these paradigms to an introduction in [24], [31], [32].

III. ML FOR NS IN THE PREPARATION PHASE

State-of-the-art NS solutions applied ML techniques for two problems in the preparation phase. First, we discuss the translation of service profiles into slice requirements. Afterward, we present the slice admission control.

A. Translation of Service Profiles Into NS Requirements

From the tenant's perspective, designing a network slice is a complex task that involves a complete description of the service topology, details on service configuration and workflows, and SLA definitions for service assurance. To make this task easier, network operators provide generic slice templates to be used as a reference by the tenants when ordering a network slice. However, some services may not have a direct mapping to a predefined slice template since service requirements may vary widely. For instance, some services may have ultra-low latency, high bandwidth, and high-reliability requirements at the same time. An alternative to this problem is to derive the slice requirements from service profiles defined through high-level intents. Despite much work on intent-driven networking [33], [34], we found only two articles addressing the intent-based design of network slices.

The first work, proposed by Gritli et al. [35], takes into consideration the set of tenant's intents, expressed as Quality of Service (QoS) requirements, and the operator's grouping policies defining the supported slice types and their QoS characteristics. The goal is to determine all slice solutions supporting the tenant's order compliant with the operator's grouping policies. To this end, the approach first maps the slice type(s) to each intent, mapping them separately to the operator's policies. It then merges these slices based on criteria such as the operator's policies they comply with and isolation and placement constraints. However, the approach presented in [35] is model-based and, thus, does not use ML. The second work, proposed by Yang et al. [25], develops a mechanism based on ML to translate service intents into a slicing configuration language. The proposed mechanism employs the Latent Dirichlet Allocation algorithm to extract keywords from an optical network topic model and construct an intent theme model. The intent issued by the users is a mixed distribution of certain topics, which is also a probability distribution of words. If the intent topic is found, the keyword in the topic is also the core meaning of this intent. To associate intent keywords with QoS constraints, the authors propose using an experienced database. The evaluation uses a discrete intent service emulator and a network topology assembled by OpenAI Gym.

B. Slice Admission Control

Over-provisioning is not possible in 5G and beyond 5G since infrastructure resources (especially spectrum) are limited. Therefore, network operators must decide which slice requests should be admitted or rejected in the infrastructure to manage resources efficiently. Specifically, the slice admission control problem is formulated as follows. Upon receiving a network slice request from a tenant, the operator's system must decide whether to accept or reject the tenant's request, pursuing a predefined objective while still honoring the agreed SLAs for previously accepted network slice requests. Such a decision is challenging as it must consider the total available system capacity, randomly arriving tenant requests, real network utilization within the already instantiated slices, and the Quality of Experience (QoE) perceived by the end-users. This section introduces recent works that applied ML to solve the slice admission control problem.

Bega et al. [36] address the problem of designing a slice admission control that maximizes the network operator revenue while satisfying the desired service guarantees. The authors consider two types of slices: elastic, which does not require any instantaneous throughput guarantees, and inelastic, which requires a certain fixed throughput to be satisfied during the entire slice life cycle. Only the RAN segment is considered in work. The slice type, slice duration, the slice size in terms of the number of users, and the price per time unit characterize a slice request. The problem is formulated as a Semi-Markov Decision Process (SMDP), where the elastic and inelastic network slice requests follow a Poisson process. Each state is modeled as a three-sized tuple representing the number of elastic and inelastic slices in the system at a given decision time and the next event (new arrival of an elastic or inelastic slice request or departure of a slice of any type) that triggers a decision process. The possible actions include admitting a new request for an elastic or inelastic slice or rejecting the new request. In the first case, the resource associated with the request is granted to the tenant, and the operator immediately earns a reward, computed as the product of the slice type price and duration. The second case has no immediate reward, but the resources remain accessible for future requests. Requests that are rejected are no longer considered by the system. The SMDP problem is then solved using Q-Learning (QL), an RL algorithm where the learning function that maps the input state to the expected reward when taking a specific action is realized as a lookup table. Simulations with the slice duration following an exponential distribution showed that QL achieved close to optimal performance.

Despite the good performance, an inherent drawback of RL algorithms such as QL is their lack of scalability when the state space becomes too large. Inspired by this limitation, in a later work, Bega et al. [37] propose a Deep Q-Learning (DQL) algorithm, named NS Neural Network Admission Control (N3AC), to solve the slice admission problem. DRL algorithms use Neural Networks (NNs) to generalize the experience learned from some states to be applied to other states with similar features. In particular, N3AC uses a feed-forward NN structure, where the neurons of one layer are fully interconnected with the neurons of the next. In addition, N3AC relies on a single hidden layer and uses the Gradient Descent approach to back-propagate the measured error at the output layer to the input layer. Furthermore, N3AC does not apply any ground truth to train the NN. This training is achieved using output estimations, which become more accurate as explorations are performed. The performance of N3AC was evaluated through simulation where the service time follows an exponential distribution and slice request arrivals follow a Poisson process.

Similar to [37], Bakri et al. [38] proposed a DQL algorithm to solve the slice admission problem. The authors compare the performance of the DQL solution with two other algorithms: QL, and Regret Matching. The QL and DQL approaches are evaluated using the offline version of the algorithms, while Regret Matching performs online. Results show that Regret Matching reacts faster to load change than the other two algorithms.

Dandachi et al. [39] propose a slice admission control considering communication, computing, and storage resources to maximize resource utilization and operator revenue. The slice admission control considers two types of slices, Best Effort (BE) and guaranteed QoS slices, with elastic requirements. Resources from the RAN and CN domains are considered. The slice admission control comprises two steps: at the beginning of each time slot, the slice admission control evaluates the similarity between the income requests and the slices already active in the system to identify slice instances that can serve the new slice requests with a minimum amount of additional resources. The first step uses a normalized spectral clustering algorithm based on the Jaccard similarity, while the second is implemented using State-Action-Reward-State-Action (SARSA). In the second step, based on the current state of system utilization, the admission control first decides whether to scale down the resources allocated to BE slice instances, then selects the income slice requests to admit. Evaluation is carried out by simulation using slice templates customized by the authors.

Reza et al. [40] propose an RL agent to decide whether or not a new slice request should be accepted. A slice request is specified in terms of its duration, service type (priority), and the number of Central Processing Unit (CPU) and link resources needed. The objective is to maximize the network operator's total revenue while matching the service requirements of the slices in operation as closely as possible. The work focuses on the RAN domain. The RL agent is implemented using a NN that receives the slice request and the resources currently available in the system as input. NN minimizes the loss of revenue derived by rejecting the slice requests and the loss derived by degrading the service of a slice in operation. NN is trained in an episodic manner, and at the end of an episode, the cumulative reward for all the actions up to the current point in time is computed. Evaluation is performed using a custom-built simulator, where the inter-arrival time of requests and slice duration are exponentially distributed.

Bakhshi et al. [41] propose a slice admission control in a federated environment formed by one consumer and provider domain. For a given slice request, the admission control decides whether to deploy the slice in the consumer or the provider domain or reject it. The decision is based on the cost of deploying the slice locally (consumer domain) or remotely (provider domain) and on the current resource availability. The model is focused on computing resources, thus more suitable for the edge, CN, and cloud domains. The authors compare the performance of two RL algorithms for solving the problem: QL and R-Learning, an average reward learning algorithm. Results are obtained through simulation using customized slice templates and show that R-Learning performs better than QL for the federated problem due to QL's dependency on the discount factor.

Sciancalepore et al. [26] propose the concept of slice overbooking, where more slice requests are admitted than the overall system capacity to maximize the operator revenue. In their proposal, a slice request comprises the amount of physical wireless resources assigned to the slice and its duration. The slice admission control problem is formulated as an online decision process using a variant of the MAB model. Each tenant is a bandit that, if pulled at a certain round, returns a particular reward. Multiple bandits can be pulled at a given round, and tenants with active slices must be selected while their slices are operational. If a lock-up period runs, the gambler must select the same arm as in the previous round. The reward accounts for the total amount of resources asked within the slice request and the ratio between what has been used and what is being asked, following the rationale that tenants underutilizing assigned resources are preferred for the ones fully using them. The authors implement the MAB model using three RL algorithms: Upper Confidence Bound (UCB), Online NETwork Slice Broker (ONETS), and ϵ-Greedy, providing a trade-off between complexity and sub-optimality. Performance evaluation is carried out by simulation. In addition, proof-of-concept implementation is presented considering three network slices: eMBB for Guaranteed Bit Rate, eMBB for BE, and Public Safety. Table II summarizes the main characteristics of the literature related to ML applied to NS problems in the preparation phase.

IV. ML for NS in the Commissioning Phase

In the commissioning phase, NS problems are mainly related to making resource allocation decisions for the admitted slices. After being admitted to the system, the slice is instantiated by allocating resources in the RAN, TN, and CN domains. A RAN slice subnet comprises the radio access and processing functions from a set of Base Stations (BSs) and the allocated Physical Resource Blocks (PRBs) to support a communication service. A CN slice subnet contains a set of network services functionalities and associated computing resources. A TN slice subnet, on the other hand, comprises a set of connections between a group of virtual or/and physical network functions from both the RAN and the CN, each one having its own SLA. This section discusses state-of-the-art ML solutions for instantiating a slice within the RAN, TN, and CN domains. First, we discuss ML resource allocation solutions for instantiating a RAN slice subnet. Then, we present ML resource allocation approaches for instantiating a TN and a CN slice subnet.

A. Radio Resource Sharing

RAN slice subnet instantiation is usually formulated as the problem where the resources of one or more BSs, i.e., spectrum, power, antennas, among others, must be shared between multiple slices [42]. In the literature, the RAN slicing problem has been tackled on two different levels: planning and runtime. In the following, we discuss works dealing with RAN slicing at the planning level. At the runtime level, RAN slicing is realized through slice elasticity, which will be discussed in Section V.

At the planning level, RAN resources are allocated to each slice before its operation based on capacity and isolation requirements. In our survey, we observed that works dealing with RAN slicing at the planning level fall into two categories: those applying a combined slice admission control and resource allocation solution and those using slice traffic/resource demand prediction. Since ML solutions for the slice admission control problem have been introduced in Section III-B, in this section, we discuss relevant works that use ML for predicting traffic/resource usage for RAN slicing.

Gutterman et al. [43] proposed a metric for a slice named REVA, defined per QoS Class Identifier (QCI) and traffic direction. REVA measures the resource rate (in PRBs/sec) available for a Very Active bearer, i.e., a bearer that continuously attempts to obtain more PRBs than a maximal fair share available. The authors then developed a prediction model for this metric and used it for slice provisioning. The work collected traces of RAN resource allocation from a custom-designed experimental Long Term Evolution (LTE) testbed under various network usage patterns to build the model prediction. The authors then designed a modified Long Short Term Memory (LSTM) model to predict REVA tens of seconds in advance. The accuracy of the LSTM was evaluated against the Autoregressive Integrated Moving Average (ARIMA) model and traditional LSTM neural networks, showing that the proposed model outperforms ARIMA and LSTM by up to 31%. Finally, the authors designed a slice provisioning algorithm that exploits the prediction models to minimize costs for service providers.

A network slice admission control coupled with resource allocation guided by a forecasting module that predicts network slices' traffic and user mobility patterns is presented by Sciancalepore et al. [44]. In their proposal, the authors assumed that traffic requests within a slice follow a periodic pattern, applying time-series forecasting based on the Holt-Winters technique to predict the aggregate traffic for every admitted slice. The authors also employed the Self-similar least-action human walk (SLAW) mobility model for user mobility prediction. Using traffic generated by this model, the authors developed a Markovian chain to capture the mobility pattern of a user and assumed that a weighted combination of such patterns reflects the mobility of a tenant. The authors

TABLE II
SUMMARY OF ML-APPROACHES FOR NS PROBLEMS IN THE PREPARATION PHASE

Ref.	NS Problem	Learning Paradigm	Learning Method	Resource Type	Network Segment	Performance Evaluation
[25]	Service profile translation	NLP	Latent Dirichlet Allocation	Network, computing, storage	TN	Emulation - Custom-built intent service
[36]	Slice admission control	RL	QL	Network	RAN	Simulation - Poisson distribution for request and exponential distribution for slice duration
[37]	Slice admission control	RL	DQL	Network	RAN	Simulation - Similar to [36]
[38]	Slice admission control	RL	QL, DQL, Regret Matching	Network	RAN	Simulation - Similar to [36]
[39]	Slice admission control	UL, RL	Norm. spectral clustering, SARSA	Network, computing, storage	RAN, CN	Simulation - Customized slice templates
[40]	Slice admission control	RL	DRL	Network, computing	RAN	Simulation - Exponential distribution for request inter-arrival time and slice duration
[41]	Slice admission control	RL	QL, R-Learning	Computing	Edge, CN, cloud	Simulation - Customized slice templates
[26]	Slice admission control	RL	UCB, ONETS, ϵ-Greedy	Network	RAN	Simulation and experimental

then employed an UL method to learn the weights of each tenant. Next, they combined the overall load predicted by the Holt-Winters method and the mobility model to derive the predicted amount of resources requested by the tenant under a BS. Finally, the authors designed a RL algorithm to perform admission control considering the SLA of the different tenants, their traffic usage, and user distribution. Performance evaluation was conducted using a MATLAB simulation with 7 BSs, 10 tenants, and 100 UEs per tenant distributed uniformly. Results show that proper forecasting increases system utilization, especially as the number of network slice requests and system capacity grows.

Sapavath et al. [45] studied the Sparse Bayesian Linear Regression (SBLR) and Support Vector Machine (SVM) techniques to estimate and predict Channel State Information (CSI) to make a decision about radio frequency slicing. The system model was composed of infrastructure providers that sublease their radio frequency for Mobile Virtual Network Operators (MVNOs) based on the requests coming from MVNOs and their SLAs. Depending on the demands and requirements, users are classified into three user groups (stationary, mobile, and indoor) and the infrastructure provider's wireless resources are allocated to MVNOs to serve the users of individual groups. Given the end-user demands, RAN resource pool, the number of available antennas, and the total bandwidth of the radio frequency slices, the solution assigns wireless resources for the slice considering the data rate of each user of the slice. This data rate, in turn, is computed based on the estimated CSI. The training dataset was acquired through pilot-based training and data augmentation. Performance evaluation focused mostly on the accuracy of the predictors and showed that SBLR results in better outcomes than SVM, demontrating that this technique is less sensitive to sparse CSI information.

B. VNF Placement

The TN and/or CN Slice Subnet Instantiation problem is usually formulated as the placement of a set of VNFs towards the underlying physical infrastructure. This approach is a typical VNE problem reformulated to consider specific requirements of the 5G system such as Random Access Memory (RAM), CPU, disk, bandwidth, and latency constraints, as well as node sharing. Indeed, in the VNF placement problem, given a physical network G, representing the underlying physical infrastructure, and a virtual network H, representing the slice, we have to embed the virtual onto the physical network so that each virtual node $m \in H$ is mapped onto a physical node in G and each virtual link $(m, n) \in H$ is mapped to a loop-free physical path in G connecting the two physical nodes to which the virtual nodes m and n have been mapped [16]. The objective is to find an embedding with the least cost that satisfies all link and node capacity constraints. The cost may represent congestion, preference in terms of operator agreements, load balancing, or real cost of operation.

The most relevant works that use ML to solve the VNF placement problem formulate it as a Markov Decision Process (MDP) and solve it using DRL. Yan et al. [46] proposed a combined DRL with a neural network structure based on graph convolutional networks to solve the VNF placement problem. In their proposal, states are represented by eight attributes: the number of CPU resources over all nodes, the amount of bandwidth available in each node, the amount of free CPU currently available in each node, the amount of bandwidth not allocated in each node, a vector describing the embedding for the current slice request, the number of CPU and bandwidth resources needed by the current slice request, and the number of unallocated virtual nodes in the current request. To reduce the number of input features, links are not explicitly considered in the state representation. Instead, a Graph Convolutional Network (GCN), a Convolutional Neural Network (CNN) used to extract features from homogeneous graphs, is employed to automatically extract link features from the physical network. The action taken by the RL agent is the index of the physical node in which to place a specific VNF of the slice. This way of modeling the actions breaks the process of placing one slice in a sequence of VNF placements and reduces the size of the action space to the number of physical nodes. The reward function combines the acceptance ratio, the placement cost, and the load balance. The solution was evaluated through simulation using a substrate network topology generated following the Waxman random graph. CPU and bandwidth resources of the substrate network were uniformly distributed between 50 and 100 units, while slice requests were generated by a Poisson process.

Rkhami et al. [47] also employed DRL and CNNs to improve the quality of a VNF placement heuristic. However, different from [46], the authors in [47] used a Relational GCN, which operates over heterogeneous graphs. The authors only consider resource-related features (CPU and bandwidth) to represent the system state, while the action is represented by a binary variable used to keep the same placement of the current VNF or to modify it based on a computed heuristic. The objective of the solution is to maximize the infrastructure provider revenue. The evaluation was performed through simulation on a network topology following the Waxman random graph. CPU and bandwidth requests are drawn uniformly, as well as the number of VNFs in each request.

A Deep Deterministic Policy Gradient (DDPG) approach is employed by Quang et al. [48]. Different from [46] and [47], in [48], the state representation includes resource-related features (CPU and bandwidth) and latency-related properties. The action taken by the DRL agent is represented by two sets of weights: one indicating the placement priority of each VNF in the slice request on each physical node and the other indicating the placement priority of each virtual link on each physical link. The reward function of action is modeled as the acceptance ratio. To assess the performance of the proposed approach, the authors employed simulations using a real-world network topology with 24 nodes and 37 links. Link capacities are randomly chosen, the requested VNF resources are uniformly distributed, and virtual links are arbitrarily requested with bandwidth in the range of 1 Mbps to 40 Mbps and latency of 1 ms to 100 ms.

Ensuring that a DRL agent converges to an optimal policy in the VNF placement problem is a challenge since its performance depends on the exploration of a huge number of states and actions. To overcome this problem, Esteves et al. [49] introduced the concept of Heuristically Assisted DRL, which combines a DRL algorithm based on Advantage Actor-Critic (A2C) and a GCN with a Power of two Choices heuristic to control the DRL convergence. The RL elements of the solution (i.e., state, action, and reward) follow the same approach in [46]. The performance evaluation is carried out through simulation with three data center types (edge, core, and cloud) and one slice type (eMBB). Slice requests involve five VNFs, and arrival rates follow three network load conditions (underload, normal load, and critical load).

Mei et al. [50] handled the VNF placement problem by creating a VNF pool. This pool integrates all individual VNFs distributed in the network domains, providing a variety of network abilities to meet the requirements of Vehicle-to-Everything (V2X) services. An Intelligent Control Layer is responsible for orchestrating the available VNF (e.g., allocating VNFs and network resources to network slices). The solution intends to support the deployment of VNFs on remote and edge clouds by using Deep Q-Network (DQN) with CNNs. The solution was evaluated through simulation with an urban scenario based on the Manhattan grid layout and two types of Vehicle-to-Vehicle (V2V) services: traffic safety and efficiency service and autonomous driving-related service.

Kibalya et al. [27] tackled the multi-domain slicing as a multi-substrate VNF problem. In their proposal, a DRL algorithm selects the optimal set of infrastructure providers among all the feasible candidates to maximize the revenue-to-cost ratio for deploying the slice requests. The DRL algorithm is based on a NN that takes as input an M x N feature matrix, where M is the number of infrastructure providers and N is the number of extracted features. The latter reflects the attributes of both the slice request and substrate network. The NN was trained offline using demands of the size of 500 requests per epoch, with the request delay uniformly distributed between 1 to 200 units. The evaluation considered an online scenario where the request arrival follows a Poison distribution. A comparison with a combinatorial scheme showed that the DRL algorithm presents a better performance, especially in the presence of high request arrival rates.

Fantacci and Picano [51] proposed an NS strategy that uses FL to support slice allocation through VNF placement in distinct service areas with different costs and processing and storage capabilities. In their proposal, UEs are mapped into three slice classes (high-rate communications; highly dynamic, low-rate, and delay-tolerant communications; and URLLC), and a FL framework is employed to foresee the UEs' demand of each service class. The goal is to use the forecast UEs' demand to provide a VNF placement that maximizes the infrastructure provider revenue while improving the end user's QoE. The FL framework applies ML models trained at the UE level, and then a central layer aggregates to improve the global learning model. To capture the UE request behavior, the authors use Prospect Theory (PT). The latter aims at evaluating a prospect (service area) defined over a set of outcomes (UE service completion time) and the probability associated with each of them. The proposed framework was evaluated through simulation involving eight different areas with processing and storage capacities, VNF types, and costs uniformly distributed. The VNFs requests were modeled by using the MovieLens dataset.

Panayiotou et al. [52] focused on the TN Slice Subnet Instantiation problem. The objective is to define a transport path considering a multi-domain network slice, which could span many paths. In this context, the authors work on the Quality of Transmission (QoT) estimation for sliceable optical networks. The authors examine centralized and distributed NN-based QoT estimation model for sliceable optical networks. The objective is to find QoT model(s) that are fine-tuned to the diverse requirements of each slice. The centralized problem is formulated as a multiclass classifier trained with global network information while the distributed problem is formulated as a set of binary classifiers, each of them trained according to data that is relevant to a single type of slice. The results show that the distributed QoT model performs better than the centralized model, being independent of the number of slice types. Table III summarizes the main characteristics of the literature related to ML applied to ML problems in the commissioning phase.

V. ML for NS in the Operation Phase

The network slice operation phase requires intense management activity in run-time. In addition to activating the network slice instance provisioned in the commissioning phase, the operation phase also cares about the supervision, performance reporting, modification, and resource capacity planning [29]. Therefore, the state-of-the-art brings several ML approaches for various network slice operation tasks. In our review, we find out that ML is often adopted to solve the following NS problems in the operation phase: network slice elasticity, user admission control; traffic classification and prediction; anomaly detection, task offloading, congestion control, RAT selection, and NS with mobility. This section details how relevant works in the state-of-the-art tackle each problem. Table IV summarizes the main characteristics of the literature related to ML applied to LCM problems in the operation phase.

A. Network Slice Elasticity

Network slice elasticity embraces run-time tasks to modify the current slice deployed to support a user demand or application requirement. Li et al. [53] brought solid contributions to reviewing the background of DRL and its usage for resource management in NS. The work follows two main scenarios: (i) resource management for RAN; and (ii) priority scheduling in typical VNF. Relying on DQL, the authors proposed an approach based on allocating resources regarding the users' activity. Such solution performed better than other intuitive approaches, such as demand prediction, no slicing, and hard slicing.

Qi et al. [54] presented an enhancement to the applicability of DQL. The authors show how to allocate/reallocate limited spectrum across slices by improving the calculation and approximation of the Q-value function. The authors argue that their approach is suitable for NS tasks, having faster convergence and better performance than typical DQL. However, they point out that there is still space for research in aspects such as SLA assurance.

Li et al. [55] proposed an algorithm for end-to-end NS resource allocation based on DQN. However, we fit this work into the network slice operation phase due to its contribution to slice elasticity, which assumes slice instantiation and execution. The authors presented a framework for 5G resource allocation, considering wireless resources on RAN and VNF on CN. A DQN algorithm uses the feedback from the environment dynamically and in real-time to update the wireless resources and map the service links. Simulations support the results in terms of access rate.

Bouzid et al. [56] demonstrated an intelligent solution for dynamic capacity allocation in an end-to-end network slice with multiple cloud-enabled virtualized segments for a video replay service. A RL algorithm is used with predictive models (trend-based and parametric methods) for state estimation. Authors argue that the predictive models with RL can manage the elasticity through a servicing gateway and Web servers and cooperate to enhance the global system efficiency.

Guan et al. [57] proposed a hierarchical resource management framework that utilizes DRL to perform resource adjustment within admitted end-to-end slices. The proposed framework introduces 1) multiple local resource manager to deal with the demand changes in resource requirements for an individual slice; and 2) a global resource manager to control the local resource managers. The local resource manager executes a DQL algorithm, where states represent the current service quality satisfaction, actions denote whether slice adaption is required, and reward is defined as the revenue obtained by adjusting resources minus the resource consumption cost and operational cost. Evaluation is performed using simulation on network and computing resources.

Indeed, because of its flexibility, dynamism, and high applicability for large-scale problems, ML techniques apply to the most diverse network slice elasticity issues, such as network performance and overall resource optimization and QoS guarantee. The vast majority of reviewed work in our survey concentrates on this category (Table IV presents a summary of all of them). In addition, most of these problems sit on the RAN segment, and DRL is the most selected ML technique to deal with them [58], [59], [60], [61], [62], [63], [64], [65], [66], [67], [68], [69], followed by supervised learning [70], [71], [72], [73], [74], [75], [76]. The authors in [30] use a Deep Neural Network (DNN) to decide on network slice reconfiguration in a Metro-Core optical network.

B. User Admission Control

The user admission control NS problem aggregates articles regarding challenges in deciding whether a new user, upon request, should be added to a running network slice or not. The difference between user admission control and slice admission control is that, in the former, the request is for including a new user into an instantiated and running network slice. Overall, user admission control is a process that ponders what is being requested *vs.* what is or will be available to be consumed (e.g., bandwidth, computing, storage, and radio spectrum). Admitting new users into a running network slice means that the operator commits to the availability of resources (e.g., spectrum and bandwidth) to serve all the users hosted in the slice.

In 5G networks, typically, user requirements may change over time (e.g., depending on the running applications) and a single UE may connect to up to 8 network slices simultaneously [77]. Therefore, ML techniques act in this decision-making based on the time-varying user requirements, possible resource allocation for newcomers, eventual slice elasticity, and long-term SLA holding, for example.

Shome and Kudeshia [78] focused on the RAN and considered three different generic slice templates: mMTC, URLLC, and eMBB. Based on a modified version of classical DQL, users are allocated/reallocated to slices regarding their current needs. In the literature, this type of problem is also called Slice Selection. However, we consider it part of user admission

TABLE III
SUMMARY OF ML-APPROACHES FOR NS PROBLEMS IN THE COMMISSIONING PHASE

Ref.	NS Problem	Learning Paradigm	Learning Method	Resource Type	Network Segment	Performance Evaluation
[43]	Radio sharing with traffic prediction	SL	Modified LSTM	Network	RAN	Experimental - Traces collected from a LTE testbed
[44]	Radio sharing with traffic prediction	Time-series, UL, and RL	Holt-Winters, Customized RL algorithm	Network	RAN	Simulation - SLAW mobility model
[45]	Radio sharing with traffic prediction	SL	SBLR, SVM	Network	RAN	Simulation - Pilot-based symbols
[46]	VNF placement	RL	GCN	Computing, network	TN, CN	Simulation - Poisson distribution for slice requests
[47]	VNF placement	RL	Relational GCN	Computing, network	TN, CN	Simulation - Uniform distribution for slice requests
[48]	VNF placement	RL	DDPG	Computing, network	TN, CN	Simulation - Uniform distribution for slice requests
[49]	VNF placement	RL	A2C, GCN, heuristic	Computing, network	Cloud, CN, Edge	Simulation - Customized arrival rate
[50]	VNF placement	RL	DQN, CNN	Computing	Cloud, Edge	Simulation - Customized V2V services
[27]	VNF placement	RL	CNN	Computing, network	TN, CN	Simulation - Poisson distribution for slice requests
[51]	VNF placement	FL	PT	Computing, storage	CN	Simulation - VNF request generated from real-world dataset
[52]	Path configuration	SL	NN	Network	TN	Simulation - Poisson distribution for connection request

since the process requires aggregating new users into running slices. In [78], the authors set up an experiment simulation scenario with multiple MVNOs sharing virtual BS resources. The experiment sets 30 MHz bandwidth for each virtual BS, distributing it among 100 users. Each one of the users has requirements fitting them in at least one of the three generic slice templates (mMTC, URLLC, eMBB). Results show that the authors' proposal keeps a high average user satisfaction score during the experiment.

Nassar and Yilmaz [79] considered a 5G scenario and discussed the limitation of resources at the network edge, specifically at fog nodes supporting vehicular and smart-city networks. The NS proposal includes creating a cluster of fog nodes with a controller, referred to as Edge Controller (EC), responsible for efficiently managing resources. The EC uses DRL to adapt to optimal slicing policies, performing admission control tasks (e.g., serving or avoiding new users, serving or avoiding specific requests) towards load balancing, saving resources, and denying tasks better performed in the cloud. The authors evaluate the proposal's suitability for the edge network using simulations.

C. Traffic Classification and Prediction

This category embraces works on the slice run-time using ML techniques for traffic classification and/or prediction. Traditionally, classifying network traffic involves three common approaches: port-based, Deep Packet Inspection (DPI), and statistical. ML techniques are especially appropriate for statistical approaches, which classify the traffic according to, for example, the packets' size and transmission direction. Therefore, the state-of-the-art presents NS methods based on classifying the traffic to infer running applications, predict bandwidth, and dynamically allocate/reallocate resources.

Le et al. [80] presented early-state contributions for the future Self-Organized Networks (SONs) NS. The authors aim to build an architecture for NS based on mobile broadband traffic classification. Based on past contributions working on big data, ML, and SDN/NFV, the authors use K-means as an UL algorithm for clustering mobile applications, resulting in three slices (0.5Mbps, 1Mbps, 3Mbps). They also apply several SL techniques (e.g., Naive Bayes, SVM, NN) for classifying new coming traffic flows into the three distinct slices.

Results show high accuracy in traffic classification and therefore promising early-state contributions.

The authors in [81] used an FL method based on Key Performance Indicator (KPI) data collection (e.g., network traffic) at virtualized Central Units (CUs) to maintain distributed local datasets, referred to as Mini-Datasets. These distributed Mini-Datasets compose the FL model for resource allocation with long-term SLA constraints. In this context, the authors have another complementary publication [82] focusing on the energy efficiency perspective of their approach.

Terra et al. [83] presented an analysis of eXplainable Artificial Intelligence (XAI) methods applied to telecommunication networks. XAI methods are applied to analyze the cause of SLA violation prediction made in 5G networks. The proposal analyzes the explanation directly generated from the SLA violation prediction instead of expert knowledge. Local Interpretable Model-Agnostic Explanations (LIME), SHapely Additive Explanations (SHAP), Permutation Importance (PI), and Extreme Gradient Boosting (XGBoost) XAI methods are used to analyze SLA violation prediction causes, and these methods are further compared among them.

Salhab et al. [84] proposed a micro-service-based experimental prototype with a regression tree algorithm to validate the impact of forecasting capabilities on the RAN

TABLE IV
SUMMARY OF ML-APPROACHES FOR NS PROBLEMS IN THE OPERATION PHASE

Ref.	NS Problem	Learning Paradigm	Learning Method	Resource Type	Network Segments	Performance Evaluation
[53]	Slice Elasticity	RL	DQL	Network	RAN, CN	Simulation
[54]	Slice Elasticity	RL	DQL	Network	RAN	Simulation
[55]	Slice Elasticity	RL	DQN	Network	RAN, TN, CN	Simulation
[56]	Slice Elasticity	RL	QL	Network	RAN, TN, CN	Not clear
[57]	Slice Elasticity	RL	DQL	Network, computing	RAN, TN, CN	Simulation
[58]	Slice Elasticity	RL	DQN	Network	RAN	Simulation
[59]	Slice Elasticity	RL	DRL	Network	RAN	Simulation
[60]	Slice Elasticity	RL	DQL	Network	RAN	Simulation
[61]	Slice Elasticity	RL	DDPG	Network	RAN	Simulation
[62]	Slice Elasticity	RL	Duel. DNN, QL	Network, computing, storage	C-RAN	Simulation
[63]	Slice Elasticity	RL	DRL	Network	RAN	Simulation
[64]	Slice Elasticity	RL	DQN	Network	RAN	Simulation
[65]	Slice Elasticity	RL	DQN	Network	RAN	Simulation
[66]	Slice Elasticity	RL	DQN	Network	RAN	Not clear
[67]	Slice Elasticity	RL	LSTM, A2C DRL	Network	RAN	Simulation
[68]	Slice Elasticity	RL	DQL	Network, computing	RAN, edge	Simulation
[69]	Slice Elasticity	RL	A2C	Network	RAN	Simulation
[70]	Slice Elasticity	SL	DNN	Network	RAN, CN	Real-world dataset
[71]	Slice Elasticity	SL	LSTM	Network	RAN, CN	Exp. & real-world dataset
[72]	Slice Elasticity	SL	LSTM	Network	RAN, TN	Real-world dataset
[73]	Slice Elasticity	SL	DNN, LSTM	Network	RAN	Simulation
[74]	Slice Elasticity	SL	DNN	Network, computing	RAN	Not clear
[75]	Slice Elasticity	SL	DNN	Network, computing	RAN, TN	Real-world dataset
[76]	Slice Elasticity	SL	LSTM	Network	RAN, TN	Experimental
[30]	Slice Elasticity	RL	DNN	Network	TN	Simulation
[78]	User Adm. Control	RL	DQL	Network	RAN	Simulation
[79]	User Adm. Control	RL	DQN	Network, computing	RAN, edge	Simulation
[80]	Traffic Prediction	UL, SL	Naive Bayes, SVM, NN	Network	RAN, CN	Experimental
[81]	Traffic Prediction	FL	Non-zero sum	Network	RAN	Real-world dataset
[82]	Traffic Prediction	XAI	XGBoost, SHAP	Network	TN	Experimental
[84]	Traffic Prediction	SL	Regression Tree	Network	RAN	Experimental
[85]	Anomaly Detection	UL, SL	HDBSCAN, DNN	Network	RAN	Simulation
[88]	Task Offloading	RL	QL	Network, computing	RAN	Simulation
[89]	Congestion Control	RL	TRPO	Network	RAN	Simulation
[90]	RAT Selection	RL	DDPG	Network	RAN	Exp. & Simulation
[91]	NS Mobility	RL	DQN	Network, computing	F-RAN	Simulation
[92]	NS Mobility	SL	DNN	Network	RAN, edge	Simulation
[93]	NS Mobility	RL	A2C, DQN	Network, computing	RAN, edge	Simulation
[94]	NS Mobility	RL	A2C, LSTM	Network	RAN	Simulation
[95]	NS Mobility	RL	QL	Network	RAN	Simulation

slicing management. The experimental prototype, based on the OpenAirInterface deployment, collects data while managing several IoT devices. This data then forms a time series used to train the regression tree. The objective is to forecast the number of PRBs to be used by each slice to dynamically provision the optimum slicing ratio out of the available pool of PRBs. Results show that the forecasting model can increase substantially the throughput of the network at the cost of increased computing resources utilization.

D. Other Investigations

This subsection groups together relevant problems for the network slice operation phase. However, in our research, no substantial amount of articles discussed an ML approach to solve them. In this sense, we present at least one publication approaching each problem. Refer to Table IV for a complete list of publications regarding the operation phase.

1) Anomaly Detection: AI-assisted anomaly detection is a classical research field in computer networks [85], [86]. Analyzing the network behavior (e.g., based on KPIs such as packet loss and downlink delay) is a management task during the network slice operation phase. In [87], the authors implemented an AI-based module for assisting administrators in detecting anomalies among services in slices deployed on a virtualized infrastructure. The solution of the authors, aiming to classify network traffic, has three phases: (i) pre-processing and feature selection, (ii) clustering, and (iii) anomaly detection. Data is modeled as a time series composed of the following features: number of lost packets per service and user, uplink and downlink delay, Reference Signal Received Power, transfer protocol, and UE received bytes. In the second phase, the time series is processed by a Hierarchical Density Based Spatial Clustering (HDBSCAN) clustering algorithm, which divides the dataset into three groups: normal, moderate, and anomalous behavior. Such clusters are used to label the samples in the time series. Finally, in the third phase, the labeled dataset is used to train a feed-forward DNN to perform a classification task. After, the DNN is used to predict anomaly and assign a cluster to new data in real-time. Preliminary results using the Network Simulator 3 (NS-3) discrete-event simulator show a high accuracy score. However, increasing the number of clusters and the algorithm granularity decreases the prediction performance.

2) Task Offloading: The task offloading problem category addresses articles regarding decision-making on the most suitable domain to run a task (e.g., UE, cloud, fog). In this sense, the authors of [88] discuss the adaptive mode selection in Fog-RAN (F-RAN), which refers to the communication mode serving each UE (e.g., Cloud-RAN (C-RAN), fog-radio access point, device-to-device).

3) Congestion Control: Selected works approaching the congestion control problem with ML fall into the scenario of connection establishment for RAN and traffic congestion control in general for 5G/6G wireless networks. The authors in [89] argue that Machine-to-Machine (M2M) network traffic may surpass Human-to-Human (H2H) in the future. However, current approaches for dealing with M2M traffic rely on legacy congestion control schemes, which will no longer suit the demand in 5G and beyond scenarios. Therefore, the authors propose an improved congestion control scheme based on RL.

4) RAT Selection: Cellular networks adopting multiple different RAT impose the well-known RAT selection challenge [96]. The article [90] presents IRIS, a shared spectrum access architecture for indoor neutral-host small cells. IRIS adopts a RL algorithm based on DDPG to dynamically price the cost of a radio spectrum block in an indoor shared environment according to the previous price, tenants (operators) demands, acquisition costs, and neutral-host revenue target.

5) NS With Mobility: This problem category considers works discussing scenarios with mobility in terms that the UE is not static. To the best of our knowledge, the main concerns, up to now (the date of this research), in the context of NS with mobility are coverage area [92]; content caching [91]; and slice migration (e.g., UE moves out of the coverage and needs reallocation to another slice) [94], [95]. Addad et al. [93] propose and evaluate two DRL-based algorithms for the intelligent selection of triggers supporting NS mobility actions. Authors argue their approach is new by considering users mobility, service mobility, and resource mobility among slices for slice, service, and resource allocation. The run-time mobility decision-making process is evaluated considering the A2C, a hybrid DRL method combining value-based and policy-based approaches and DQN.

VI. Open Research Issues and Future Directions

This section identifies and discusses a non-exhaustive set of open issues on ML for intelligent NS. The identified challenges result from our analysis of the preparation, commissioning, and operation phases of the NS process, presented in Sections III–V. Moreover, we highlight the main gaps in the literature between requirements and proposals for intelligent NS.

A. Intelligent Translation for NS Requirements

Translation of service profiles into NS requirements is a complex task that requires low-level network slice configuration parameters, such as virtual machine parameters, network configurations, topology, and protocols [97]. With the evolution of networks toward beyond 5G, the complexity of this task tends to increase [98]. Consequently, an intent layer will be required to translate service profiles into slice requirements [99].

Intent-driven networking was conceived to enable applications to express desired operational goals using high-level descriptive specifications known as intents [100]. Addressing this goal, however, poses several challenges, among them, defining rich semantics to express the intent of verticals [101]. Although the integration with AI technologies, and Natural Language Processing (NLP) in particular, can bridge this gap, those technologies are still at their early stage and require further research efforts before being integrated into the network slice LCM [99]. This research gap can be evinced in our survey, where only one work [25] uses ML to bridge this gap.

B. Datasets and Experiment Reproduction

High-quality datasets are essential to support the extensive dissemination of ML in various application domains. Intelligent NS, according to our research survey, is yet another area where openly available high-quality datasets are a research issue, regardless of the slice life cycle phase, as can be seen in the column Performance Evolution of Tables II, III, and IV. A directly related aspect of dataset availability is experiment reproduction. In effect, the unavailability of datasets for most of the research work is an obstacle to allowing experiment reproduction and, to some extent, the explainability of the proposed solutions and their dissemination.

In our survey, most works (e.g., in [36], [37], [38], [40], [46] [27], [47], [48], [49], [52], [53], [54], [55], [57], among others) use data generated from simulation to evaluate their ML solutions. However, to evaluate the effectiveness of ML approaches when dealing with NS problems in practice, extensive evaluations are needed taking more realistic scenarios into consideration. To this end, some works [26], [43], [71], [76], [80], [83], [84], [90] create specific experimental testbeds for validating their model or algorithm. Although such initiatives are important, data collected from testbeds still misses the representative of the complexity and dynamicity of real-world mobile networks [102]. In addition, none of such works have made the collected data available for the research community, hindering and compromising the reproduction of the experimental parts deployed for validation purposes. Finally, 10% of the surveyed works [51], [70], [71], [72], [75], [81] use real-world network data. Although such data are much richer and more representative than those generated from simulation or testbeds, they still may suffer from noise, sparsity, and lack of label, which limits the ML algorithms that can be applied. In summary, rich and adequate data is still an issue for applying ML in NS problems.

C. Suitability of the ML Technique for the Network Slice Life Cycle Phase

While ML is an unquestionable enable for the realization of NS, it is impossible to find a single technique that completely addresses the requirements of all the network slice LCM problems. Thus, an open research issue in ML-enhanced NS is the suitability of the ML techniques for the target network slice life cycle phase with regard to, for example, granularity or timing [23]. In the preparation phase, as the slice does not exist, ML techniques using offline learning can be applied to solve the problems of such phase. Indeed, the authors in [38] conclude that offline training solutions for the slice admission control problem require a training period before use but give the best results.

In the slice commissioning and operation phases, SL, which usually relies on offline learning, has been usually applied to solve traffic prediction, traffic classification, and anomaly detection problems [43], [45], [80], [84], [87]. However, problems that involve resource allocation, such as radio resource sharing, VNF placement, and network slice elasticity, have to make decisions on low scale and cannot afford for a period of training time [103]. Thus, some works [44], [56], [88], [89] use classical RL algorithms with online training for these problems. However, resource management in NS usually involves multidimensional parameters, leading to a large state space and a low convergence rate to the optimal policy [103]. In practice, this means that, until the RL algorithm converges, it can make bad resource management decisions. Although DRL algorithms have been used to face this limitation (e.g., in [27], [46], [47], [48], [49], [50], [53], [54], [55], [57], [79], among others), DRL solutions present some shortcomings. First, DRL algorithms usually rely on DQN to encode state. However, an important component of DQN is a target network, i.e., a copy of the estimated value function that is kept fixed for some number of steps to stabilize learning [104], [105]. This copy, in turn, prevents the algorithm from reacting fast to environment changes, a desired property of RL. More recently, other NN algorithms (e.g., in [67]) have been investigated to deal with this problem. Nevertheless, further investigations are required to determine their efficacy and generalization in the context of DRL. With this regard, TL has also been considered a possible solution [103]. Another problem with DRL algorithms is that NNs with multiple layers cannot explain the essential features that influence their decisions or the impact of data bias on the uncertainty of outputs [106]. As network slices are expected to host an increasing number of mission-critical services in beyond 5G, trust will become critical. Despite this need, our survey identified only one work [83] addressing explainable ML-enhanced NS. Finally, it is important to highlight that DRL algorithms have a high demand for computing, memory, and energy resources [102]. Considering that beyond 5G networks will make pervasive use of intelligence [21], DRL algorithms that make more efficient use of computing and energy resources are still an open issue.

D. End-To-End NS

NS is applied in challenging systems such as 5G and beyond 5G, Industry 4.0/5.0, and intelligent transportation systems. End-to-end NS is an essential requirement and current trend for these systems. However, in most surveyed works, ML support is focused on network segment solutions (e.g., RAN [88], RAN + edge [79], TN [83], and RAN + CN [70], leaving end-to-end NS as an open research issue. We consider that an article effectively approaches end-to-end NS if it deals with the three network segments (RAN, TN, and CN) completely. However, this issue is not a consensus in the literature. For example, in [76], the authors assume that the end-to-end can start inside the RAN, crosses a TN, and finishes at the frontier of a CN. The authors do not consider the fronthaul, i.e., part of the RAN is not sliced, nor the CN. In our survey, only a few works effectively tackle the end-to-end slicing problem in the three segments (RAN, TN, and CN) [55], [56], [57], and the first two only deal with network resources. While [57] is a more comprehensive work considering network and computing resources, the performance evaluation in this article is based on a small and simplified simulation. The authors are focused on calling attention to the importance of ML-enabled NS in 6G and the challenges in the real-world implementation.

Slicing by segment with ML support is undoubtedly relevant. Nevertheless, the end-to-end design must consider the interdependence of resource allocation and orchestration among network segments. End-to-end intelligent slicing brings another level of complexity, which involves issues such as the need for a high-efficient (re)learning process and coordination among multiple entities [57]. In this context, FL and other distributed learning approaches may be relevant since the works can explore the spread processing capacity offered by edge computing and reduce the amount of information exchange.

E. Open RAN Intelligent Slicing

RANs are a fundamental part of the slicing process in 5G networks and Open RAN (O-RAN) is one the most relevant evolution aspect towards 6G in this segment. Not surprisingly, O-RAN architecture has AI and ML workflows in its design [107]. The O-RAN approach brings a new level of flexibility for network operators allowing them to deploy the RAN segment, potentially focusing on the business. NS has being considered a very important capability in the O-RAN context and has been already investigated in some articles [108], [109], [110]. O-RAN slicing with ML allows efficient RAN deployment to accomplish challenging user requirements regarding SLA, QoE, and user mobility.

The design, implementation, deployment, and evaluation of O-RAN with ML is a hot research topic and open research issue. In the context of O-RAN, an ML-based solution must be designed and implemented as a xApp and/or a rApp, depending on their time demands. While xApps run over a near-real-time RAN Intelligent Controller (RIC) (10ms to 1s), rApps run over a non-real-time RIC (more than 1s) [107]. Deployment and evaluation of xApps and/or rApps still depend on simulation (e.g., [108]), previously collected (and so non-interactive) datasets (e.g., [109]), or limited-size testbeds using early-stage RICs (e.g., [110]). In fact, even the optimized deployment and operation of the RICs components are challenging since they are a new software platform still under development.

F. From Theory to Practice

Based on the literature presented in the previous sections, it is clear that several theoretical works are using AI and ML,

TABLE V
SUMMARY OF ACRONYMS

Acronym	Definition	Acronym	Definition
3GPP	3rd Generation Partnership Project	N3AC	Neural Network Admission Control
5G	fifth-generation	NFV	Network Function Virtualization
6G	sixth-generation	NN	Neural Network
A2C	Advantage Actor-Critic	NS	Network Slicing
ACM	Association for Computing Machinery	NS-3	Network Simulator 3
AI	Artificial Intelligence	NSS	Network Slice Subnet
ARIMA	Autoregressive Integrated Moving Average	O-RAN	Open RAN
BE	Best Effort	ONETS	Online NETwork Slice Broker
BS	Base Station	PI	Permutation Importance
CN	Core Network	PRB	Physical Resource Block
CNN	Convolutional Neural Network	PT	Prospect Theory
CPU	Central Processing Unit	QCI	QoS Class Identifier
C-RAN	Cloud-RAN	QoE	Quality of Experience
CSI	Channel State Information	QoS	Quality of Service
CU	Central Unit	QoT	Quality of Transmission
DDPG	Deep Deterministic Policy Gradient	RAM	Random Access Memory
DQL	Deep Q-Learning	RAN	Radio Access Network
DQN	Deep Q-Network	RAT	Radio Access Technology
DL	Deep Learning	RIC	RAN Intelligent Controller
DNN	Deep Neural Network	RL	Reinforcement Learning
DPI	Deep Packet Inspection	RU	Radio Unit
DRL	Deep Reinforcement Learning	SARSA	State-Action-Reward-State-Action
DU	Distributed Unit	SDN	Software-Defined Network
EC	Edge Controller	SDO	Standards Developing Organization
FL	Federated Learning	SHAP	SHapely Additive Explanations
F-RAN	Fog-Radio Access Network	SL	Supervised Learning
GCN	Graph Convolutional Network	SLA	Service Level Agreement
H2H	Human-to-Human	SLAW	Self-similar least-action human walk
HDBSCAN	Hierarchical Density Based Spatial Clustering	SMDP	Semi-Markov Decision Process
IEEE	Institute of Electrical and Electronics Engineers	SMO	Service Management and Orchestration
IoT	Internet of IoT	SON	Self-Organized Network
IP	Internet Protocol	SVM	Support Vector Machine
KPI	Key Performance Indicator	TL	Transfer Learning
LCM	Life Cycle Management	TN	Transport Network
NLP	Natural Language Processing	TRPO	Trust Region Policy Optimization
LIME	Local Interpretable Model-Agnostic Explanations	UCB	Upper Confidence Bound
LSTM	Long Short Term Memory	UE	User Equipment
LTE	Long Term Evolution	UL	Unsupervised Learning
M2M	Machine-to-Machine	URLLC	ultra-Reliable and Low-Latency Communications
MAB	Multi-Armed Bandit	V2V	Vehicle-to-Vehicle
MDP	Markov Decision Process	V2X	Vehicle-to-Everything
ML	Machine Learning	VNE	Virtual Network Embedding
mMTC	Machine Type Communication	VNF	Virtualized Network Function
MPLS	Multiprotocol Label Switching	XAI	eXplainable Artificial Intelligence
MVNO	Mobile Virtual Network Operator	XGBoost	Extreme Gradient Boosting
ZSM	Zero touch network & Service Management		

considering the life cycle phases of NS. However, scientific research with practical and experimental approaches to NS is still in the beginning. As mentioned previously, most works use only simulation for validating their proposals, while some works focus on real-world traces or datasets, which is very useful for ML-based approaches. However, they also face hard issues such as information from outdated pre-NS technologies (LTE/4G, for instance) or data with low statistical relevance.

As expected, only few works [26], [43], [71], [76], [80], [83], [84], [90] have already accepted the challenge of evaluating an ML-based approach in an experimental testbed. In this context, not only scientific but also technological issues become relevant. For example, technological advances in telecommunications are increasingly based on native cloud computing platforms. Nevertheless, these platforms were not designed to support telecommunication services natively. Indeed, in our survey, only the work [84] has validated its proposal using a micro-service-based RAN experimental prototype.

Pushing the frontier of science by integrating theoretical advances in AI and ML with practical solutions for NS is an open issue that needs further investigation and development efforts.

VII. Final Considerations

This survey focused on presenting NS with ML research contributions. The contributions are organized by the phases of the slice life cycle as defined by standardization organizations (preparation, commissioning, and operation phases), aiming to identify trends and correlated contributions for the different slicing phases. The contributions are concentrated in the 5G domain, with few NS solutions applied to other areas. Specifically, in the 5G domain, the end-to-end solution is a trend not yet fully explored, and ML is being extensively used to provide intelligence for segmented solutions. 5G end-to-end NS approach allows a global view of the resource allocation problem allowing for optimizing resource sharing aiming, for instance, to improve operation, achieve efficient management, and optimize operational expenditure. Although 5G end-to-end slicing is essential for service providers and telecommunications operators, the surveyed articles primarily focus on slicing and optimizing segments like RAN and the CN.

We have observed that ML is already being investigated to solve several tasks in slice preparation, commissioning, and operation. In this context, different ML techniques and algorithms have been employed, mainly the ones popularized in the last decades, such as CNN, GCN, LSTM, DRL, and XGBoost. ML has exhibited satisfactory or promising results in many automation tasks in the slice life cycle, which is critical to provide many benefits related to the concept of Zero touch network & Service Management (ZSM). However, the practical and wide adoption of ML-enabled NS still faces several challenges. Some of these challenges, such as large and open datasets and the explainability of ML-based solutions, are already being tackled by the academy and industry, which can count on the experience from other areas such as computer vision and natural language processing. However, other issues, such as the demand for short-time for model training, energy-efficient ML solutions, and distributed computation of ML models, still need much investigation. AI and ML are also evolving intensely, giving rise to new models, algorithms, techniques, and even hardware architectures. Traditionally, these novelties are not designed or tested first in networking. However, they must be imported and sometimes adapted in NS, for example.

Finally, we highlight the availability of various multi-technology (SDN, wireless, IoT, slicing, and others) testbeds worldwide for experimental research development. These testbeds, in most cases, inherently facilitate experiment reproduction using openly available software to control the experiment and having the ability to create experimental datasets. An essential point for researchers would be to evaluate to what extent these testbeds can be used for developing and validating research results in intelligent NS.

REFERENCES

[1] ITU-T. "New services and capabilities for network 2030: Description, technical gap and performance target analysis." 2019. Accessed: Jun. 20, 2022. [Online]. Available: https://www.itu.int/dms_pub/itu-t/opb/fg/T-FG-NET2030--2019-SUB.G2-PDF-E.pdf

[2] GSMA. "E2E network slicing architecture." 2021. Accessed: Jun. 20, 2022. [Online]. Available: https://www.gsma.com/newsroom/wp-content/uploads/NG.127--v1.0--2.pdf

[3] ITU. "IMT vision-framework and overall objectives of the future development of IMT for 2020 and beyond." 2016. Accessed: Mar. 25, 2022. [Online]. Available: https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.2083--0-201509-I!!PDF-E.pdf

[4] NGMN Alliance. "Description of network slicing concept." 2016. Accessed: Dec. 27, 2021. [Online]. Available: https://ngmn.org/wp-content/uploads/160113_NGMN_Network_Slicing_v1_0.pdf

[5] Ordonez-Lucena, P. Ameigeiras, D. Lopez, J. J. Ramos-Munoz, J. Lorca, and J. Folgueira, "Network slicing for 5G with SDN/NFV: Concepts, architectures, and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 80–87, May 2017.

[6] 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.

[7] X. Shen et al., "AI-assisted network-slicing based next-generation wireless networks," *IEEE Open J. Veh. Technol.*, vol. 1, no. 1, pp. 45–66, Jan. 2020, doi: 10.1109/OJVT.2020.2965100.

[8] 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.

[9] A. Kaloxylos, "A survey and an analysis of network slicing in 5G networks," *IEEE Commun. Stand. Mag.*, vol. 2, no. 1, pp. 60–65, Mar. 2018.

[10] S. Zhang, "An overview of network slicing for 5G," *IEEE Wireless Commun.*, vol. 26, no. 3, pp. 111–117, Jun. 2019.

[11] 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.

[12] 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.

[13] M. Chahbar, G. Diaz, A. Dandoush, C. Cérin, and K. Ghoumid, "A comprehensive survey on the E2E 5G network slicing model," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 49–62, Mar. 2021.

[14] J. Ordonez-Lucena, P. Ameigeiras, L. M. Contreras, J. Folgueira, and D. R López, "On the Rollout of network slicing in carrier networks: A technology radar," *Sensors*, vol. 21, no. 23, p. 8094, 2021.

[15] 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.

[16] S. Vassilaras et al., "The algorithmic aspects of network slicing," *IEEE Commun. Mag.*, vol. 55, no. 8, pp. 112–119, Aug. 2017.

[17] R. Su et al., "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.

[18] 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.

[19] 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.

[20] 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. 5G Future (ITU K)*, 2018, pp. 1–8.

[21] W. Wu et al., "AI-native network slicing for 6G networks," *IEEE Wireless Commun.*, vol. 29, no. 1, pp. 96–103, Feb. 2022.

[22] D. Bega, M. Gramaglia, R. Perez, M. Fiore, A. Banchs, and X. Costa-Prez, "AI-based autonomous control, management, and orchestration in 5G: From standards to algorithms," *IEEE Netw.*, vol. 34, no. 6, pp. 14–20, Nov./Dec. 2020.

[23] R. Li et al., "Intelligent 5G: When cellular networks meet artificial intelligence," *IEEE Wireless Commun.*, vol. 24, no. 5, pp. 175–183, Oct. 2017.

[24] R. Boutaba et al., "A comprehensive survey on machine learning for networking: Evolution, applications and research opportunities," *J. Internet Services Appl.*, vol. 9, no. 1, p. 16, 2018.

[25] H. Yang, K. Zhan, B. Bao, Q. Yao, J. Zhang, and M. Cheriet, "Automatic guarantee scheme for intent-driven network slicing and reconfiguration," *J. Netw. Comput. Appl.*, vol. 190, Sep. 2021, Art. no. 103163.

[26] V. Sciancalepore, L. Zanzi, X. Costa-Pérez, and A. Capone, "ONETS: Online network slice broker from theory to practice," *IEEE Trans. Wireless Commun.*, vol. 21, no. 1, pp. 121–134, Jan. 2022.

[27] 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.

[28] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Pérez, "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.

[29] 3GPP. "Management and orchestration; concepts, use cases and requirements (release 17)." 2021. Accessed: Jun. 20, 2022. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/28_series/28.530/
[30] S. Troia, R. Alvizu, and G. Maier, "Reinforcement learning for service function chain reconfiguration in NFV-SDN metro-core optical networks," *IEEE Access*, vol. 7, pp. 167944–167957, 2019.
[31] C. Jiang, H. Zhang, Y. Ren, Z. Han, K.-C. Chen, and L. Hanzo, "Machine learning paradigms for next-generation wireless networks," *IEEE Wireless Commun.*, vol. 24, no. 2, pp. 98–105, Apr. 2017.
[32] B. Mao, F. Tang, Y. Kawamoto, and N. Kato, "AI models for green communications towards 6G," *IEEE Commun. Surveys Tuts.*, vol. 24, no. 1, pp. 210–247, 1st Quart., 2022.
[33] E. Zeydan and Y. Turk, "Recent advances in intent-based networking: A survey," in *Proc. IEEE 91st Veh. Technol. Conf. (VTC2020-Spring)*, 2020, pp. 1–5.
[34] L. Velasco et al., "End-to-end intent-based networking," *IEEE Commun. Mag.*, vol. 59, no. 10, pp. 106–112, Oct. 2021.
[35] N. Gritli, F. Khendek, and M. Toeroe, "Network slice provisioning taking into account tenant intents and operator policies," in *Proc. IEEE Conf. Netw. Funct. Virtualization Softw. Defined Netw. (NFV-SDN)*, 2021, pp. 167–173.
[36] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, K. Samdanis, and X. P. Costa, "Optimising 5G infrastructure markets: The business of network slicing," in *Proc. IEEE Conf. Comput. Commun.*, 2017, pp. 1–9.
[37] 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.
[38] S. Bakri, B. Brik, and A. Ksentini, "On using reinforcement learning for network slice admission control in 5G: Offline vs. Online," *Int. J. Commun. Syst.*, vol. 34, no. 7, 2021, Art. no. e4757.
[39] 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.
[40] M. R. Raza, C. Natalino, P. Öhlen, L. Wosinska, and P. Monti, "Reinforcement learning for slicing in a 5G flexible RAN," *J. Lightw. Technol.*, vol. 37, no. 20, pp. 5161–5169, 2019.
[41] B. Bakhshi, J. Mangues-Bafalluy, and J. Baranda, "R-learning-based admission control for service federation in multi-domain 5G networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2021, pp. 1–6.
[42] P. L. Vo, M. N. H. Nguyen, T. A. Le, and N. H. Tran, "Slicing the edge: Resource allocation for RAN network slicing," *IEEE Wireless Commun. Lett.*, vol. 7, no. 6, pp. 970–973, Dec. 2018.
[43] C. Gutterman, E. Grinshpun, S. Sharma, and G. Zussman, "RAN resource usage prediction for a 5G slice broker," in *Proc. 20th ACM Int. Symp. Mobile Ad Hoc Netw. Comput.*, 2019, pp. 231–240.
[44] V. Sciancalepore, X. Costa-Perez, and A. Banchs, "RL-NSB: Reinforcement learning-based 5G network slice broker," *IEEE/ACM Trans. Netw.*, vol. 27, no. 4, pp. 1543–1557, Aug. 2019.
[45] N. N. Sapavath, D. B. Rawat, and M. Song, "Machine learning for RF slicing using CSI prediction in software defined large-scale MIMO wireless networks," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 4, pp. 2137–2144, Oct.-Dec. 2020.
[46] Z. Yan, J. Ge, Y. Wu, L. Li, and T. Li, "Automatic virtual network embedding: A deep reinforcement learning approach with graph convolutional networks," *IEEE J. Sel. Areas Commun.*, vol. 38, no. 6, pp. 1040–1057, Jun. 2020.
[47] 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)*, 2021, pp. 1–6.
[48] P. T. A. Quang, Y. Hadjadj-Aoul, and A. Outtagarts, "A deep reinforcement learning approach for VNF forwarding graph embedding," *IEEE Trans. Netw. Service Manag.*, vol. 16, no. 4, pp. 1318–1331, Dec. 2019.
[49] J. J. A. Esteves, A. Boubendir, F. Guillemin, and P. Sens, "A heuristically assisted deep reinforcement learning approach for network slice placement," *IEEE Trans. Netw. Service Manag.*, vol. 19, no. 4, pp. 4794–4806, Dec. 2022.
[50] J. Mei, X. Wang, and K. Zheng, "Intelligent network slicing for V2X services toward 5G," *IEEE Netw.*, vol. 33, no. 6, pp. 196–204, Nov./Dec. 2019.
[51] R. Fantacci and B. Picano, "When network slicing meets prospect theory: A service provider revenue Maximization framework," *IEEE Trans. Veh. Technol.*, vol. 69, no. 3, pp. 3179–3189, Mar. 2020.
[52] T. Panayiotou, G. Savva, I. Tomkos, G. Ellinas, "Centralized and distributed machine learning-based QoT estimation for sliceable optical networks," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, 2019, pp. 1–7.
[53] R. Li et al., "Deep reinforcement learning for resource management in network slicing," *IEEE Access*, vol. 6, pp. 74429–74441, 2018.
[54] 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.
[55] 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.
[56] M. Bouzid et al., "Cooperative AI-based E2E network slice scaling," in *Proc. IEEE Conf. Comput. Commun. Workshops (INFOCOM WKSHPS)*, Apr. 2019, pp. 959–960.
[57] 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./Oct. 2021.
[58] G. Sun, K. Xiong, G. O. Boateng, D. Ayepah-Mensah, G. Liu, and W. Jiang, "Autonomous resource provisioning and resource customization for mixed traffics in Virtualized radio access network," *IEEE Syst. J.*, vol. 13, no. 3, pp. 2454–2465, Sep. 2019.
[59] A. Aslan, G. Bal, and C. Toker, "Dynamic resource management in next generation networks with dense user traffic," in *Proc. IEEE Int. Black Sea Conf. Commun. Netw. (BlackSeaCom)*, 2020, pp. 1–6.
[60] 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.*, 2020, pp. 1–6.
[61] 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.*, 2020, pp. 1–6.
[62] 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.*, 2020, pp. 1–6.
[63] Y. Abiko, T. Saito, D. Ikeda, K. Ohta, T. Mizuno, and H. Mineno, "Radio resource allocation method for network slicing using deep reinforcement learning," in *Proc. Int. Conf. Inf. Netw. (ICOIN)*, 2020, pp. 420–425.
[64] 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.*, 2020, pp. 1–7.
[65] 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.
[66] I. Vilà, J. Pérez-Romero, O. Sallent, and A. Umbert, "A multi-agent reinforcement learning approach for capacity sharing in multi-tenant scenarios," *IEEE Trans. Veh. Technol.*, vol. 70, no. 9, pp. 9450–9465, Sep. 2021.
[67] J. Mei, X. Wang, and K. Zheng, "Semi-Decentralized network slicing for reliable V2V service provisioning: A model-free deep reinforcement learning approach," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 8, pp. 12108–12120, Aug. 2022.
[68] 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)*, 2019, pp. 1–6.
[69] Y. Xu et al., "Constrained reinforcement learning for resource allocation in network slicing," *IEEE Commun. Lett.*, vol. 25, no. 5, pp. 1554–1558, May 2021.
[70] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Prez, "AZTEC: Anticipatory capacity allocation for zero-touch network slicing," in *Proc. IEEE Conf. Comput. Commun.*, 2020, pp. 794–803.
[71] K. Abbas, T. A. Khan, M. Afaq, and W.-C. Song, "Network slice Lifecycle management for 5G mobile networks: An intent-based networking approach," *IEEE Access*, vol. 9, pp. 80128–80146, 2021.
[72] C. Zhang, M. Fiore, and P. Patras, "Multi-service mobile traffic forecasting via convolutional long short-term memories," in *Proc. IEEE Int. Symp. Meas. Netw. (M & N)*, 2019, pp. 1–6.
[73] 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)*, 2020, pp. 286–296.
[74] X. Li et al., "5Growth: An end-to-end service platform for automated deployment and management of vertical services over 5G networks," *IEEE Commun. Mag.*, vol. 59, no. 3, pp. 84–90, Mar. 2021.

[75] H. Chergui and C. Verikoukis, "OPEX-limited 5G RAN slicing: An over-Dataset constrained deep learning approach," in *Proc. IEEE Int. Conf. Commun. (ICC)*, 2020, pp. 1–6.

[76] 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. (ICC)*, 2020, pp. 1–7.

[77] 3GPP. "Management and orchestration; concepts, use cases and requirements (release 16)." 2021. Accessed: 20 Jun. 20, 2022. [Online]. Available: https://www.3gpp.org/ftp/Specs/archive/21_series/21.916/

[78] 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.

[79] 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.

[80] L.-V. Le, B.-S. Paul Lin, L.-P. Tung, and D. Sinh, "SDN/NFV, machine learning, and big data driven network slicing for 5G," in *Proc. IEEE 5G World Forum (5GWF)*, 2018, pp. 20–25.

[81] H. Chergui, L. Blanco, and C. Verikoukis, "CDF-aware federated learning for low SLA violations in beyond 5G network slicing," in *Proc. IEEE Int. Conf. Commun.*, 2021, pp. 1–6.

[82] H. Chergui et al., "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.

[83] A. Terra, R. Inam, S. Baskaran, P. Batista, I. Burdick, and E. Fersman, "Explainability methods for identifying root-cause of SLA violation prediction in 5G network," in *Proc. IEEE Global Commun. Conf.*, Dec. 2020, pp. 1–7.

[84] N. Salhab, R. Langar, R. Rahim, S. Cherrier, and A. Outtagarts, "Autonomous network slicing prototype using machine-learning-based forecasting for radio resources," *IEEE Commun. Mag.*, vol. 59, no. 6, pp. 73–79, Jun. 2021.

[85] Y. Sani, A. Mohamedou, K. Ali, A. Farjamfar, M. Azman, and S. Shamsuddin, "An overview of neural networks use in anomaly intrusion detection systems," in *Proc. IEEE Student Conf. Res. Develop. (SCOReD)*, 2009, pp. 89–92.

[86] K. Abdelli, J. Y. Cho, F. Azendorf, H. Griesser, C. Tropschug, and S. Pachnicke., "Machine-learning-based anomaly detection in optical fiber monitoring," *J. Opt. Commun. Netw.*, vol. 14, no. 5, pp. 365–375, 2022.

[87] V. A. Cunha et al., "5 Growth: Secure and reliable network slicing for verticals," in *Proc. Joint Eur. Conf. Netw. Commun. 6G Summit (EuCNC/6G Summit)*, 2021, pp. 347–352.

[88] 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.

[89] A. Turan, M. Köseoglu, and E. A. Sezer, "Reinforcement learning based adaptive access class barring for RAN slicing," in *Proc. IEEE Int. Conf. Commun. Workshops (ICC Workshops)*, 2021, pp. 1–6.

[90] X. Foukas, M. K. Marina, and K. Kontovasilis, "Iris: Deep reinforcement learning driven shared spectrum access architecture for indoor neutral-host small cells," *IEEE J. Sel. Areas Commun.*, vol. 37, no. 8, pp. 1820–1837, Aug. 2019.

[91] 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.

[92] P. Yang, X. Xi, K. Guo, T. Q. S. Quek, J. Chen, and X. Cao, "Proactive UAV network slicing for URLLC and mobile broadband service multiplexing," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 10, pp. 3225–3244, Oct. 2021.

[93] R. A. Addad, D. L. Cadette Dutra, T. Taleb, and H. Flinck, "Toward using reinforcement learning for trigger selection in network slice mobility," *IEEE J. Sel. Areas Commun.*, vol. 39, no. 7, pp. 2241–2253, Jul. 2021.

[94] 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.

[95] Y. Sun, G. Feng, L. Zhang, P. V. Klaine, M. A. Imran, and Y.-C. Liang, "Distributed learning based handoff mechanism for radio access network slicing with data sharing," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2019, pp. 1–6.

[96] L. Wu and K. Sandrasegaran, "A survey on common radio resource management," in *Proc. 2nd Int. Conf. Wireless Broadband Ultra Wideband Commun. (AusWireless)*, 2007, p. 66.

[97] C. Casetti et al., "Network slices for vertical industries," in *Proc. IEEE Wireless Commun. Netw. Conf. Workshops (WCNCW)*, 2018, pp. 254–259.

[98] Y. Wei, M. Peng, and Y. Liu, "Intent-based networks for 6G: Insights and challenges," *Digit. Commun. Netw.*, vol. 6, no. 3, pp. 270–280, 2020.

[99] B. Martini, M. Gharbaoui, and P. Castoldi, "Intent-based network slicing for SDN vertical services with assurance: Context, design and preliminary experiments," *Future Gener. Comput. Syst.*, vol. 142, pp. 101–116, May 2023.

[100] A. Clemm, L. Ciavaglia, L. Z. Granville, and J. Tantsura. "Intent-based networking-concepts and overview." 2022. Accessed: Apr. 16, 2022. [Online]. Available: https://datatracker.ietf.org/doc/rfc9315/,

[101] L. Pang, C. Yang, D. Chen, Y. Song, and M. Guizani, "A survey on intent-driven networks," *IEEE Access*, vol. 8, pp. 22862–22873, 2020.

[102] C. Benzaid and T. Taleb, "AI-driven zero touch network and service management in 5G and beyond: Challenges and research directions," *IEEE Netw.*, vol. 34, no. 2, pp. 186–194, Mar./Apr. 2020.

[103] S. Fu, F. Yang, and Y. Xiao, "AI inspired intelligent resource management in future wireless network," *IEEE Access*, vol. 8, pp. 22425–22433, 2020.

[104] S. Kim, K. Asadi, M. Littman, and G. Konidaris, "DeepMellow: Removing the need for a target network in deep Q-learning," in *Proc. IJCAI*, 2019, pp. 2733–2739.

[105] N. Saha, M. Zangooei, M. Golkarifard, and R. Boutaba, "Deep reinforcement learning approaches to network slice scaling and placement: A survey," *IEEE Commun. Mag.*, vol. 61, no. 2, pp. 82–87, Feb. 2023.

[106] W. Guo, "Explainable artificial intelligence for 6G: Improving trust between human and machine," *IEEE Commun. Mag.*, vol. 58, no. 6, pp. 39–45, Jun. 2020.

[107] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 2, pp. 1376–1411, 2nd Quart., 2023.

[108] M. Karbalaee Motalleb, V. Shah-Mansouri, S. Parsaeefard, and O. L. Alcaraz López, "Resource allocation in an open RAN system using network slicing," *IEEE Trans. Netw. Service Manag.*, vol. 20, no. 1, pp. 471–485, Mar. 2023.

[109] J. Thaliath et al., "Predictive closed-loop service automation in O-RAN based network slicing," *IEEE Commun. Stand. Mag.*, vol. 6, no. 3, pp. 8–14, Sep. 2022.

[110] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "ColO-RAN: Developing machine learning-based xApps for open RAN closed-loop control on programmable experimental platforms," *IEEE Trans. Mobile Comput.*, early access, Jul. 4, 2022, doi: 10.1109/TMC.2022.3188013.

Adnei Donatti received the Bachelor of Technology degree and the M.Sc. degree in applied computing from Santa Catarina State University (UDESC). He is currently pursuing the Ph.D. degree with the University of São Paulo working with resource allocation for network slicing on the SFI2 project. He is interested in cloud computing, distributed systems, and computer networks.

Sand L. Corrêa received the bachelor's degree in computer science from the Federal University of Goiás (UFG) in 1994, the M.Sc. degree in computer science from the State University of Campinas in 1997, and the D.Sc. degree in informatics from the Pontifical Catholic University of Rio de Janeiro in 2011. Since 2010, she has been an Associate Professor with the Institute of Informatics, UFG.

Joberto S. B. Martins (Life Senior Member, IEEE) received the Ph.D. degree in computer science from the Université Pierre et Marie Curie, Paris, in 1986. He was a Postdoctoral Fellow with ICSI, Berkeley University in 1995 and a Postdoctoral Senior Researcher with Paris Saclay University, France, in 2016. He has been an International Professor with Hochschule für Technik und Wirtschaft des Saarlandes, Germany, since 2004 and Université d'Evry, France. He is a Full Professor of Computer Science with Salvador University (UNIFACS) and the Director of NUPERC and IPQoS Research Groups with research interests in network slicing, resource allocation, machine learning, SDN-OpenFlow, Internet of Things, smart grid, and smart city. He is a key speaker, teacher, and invited lecturer at various international congresses and companies in Brazil, the U.S., and Europe.

Antonio J. G. Abelem (Member, IEEE) received the Ph.D. degree in computer science from the Catholic University of Rio de Janeiro (PUC-Rio) in 2003. He is a Full Professor with the Computer Science Faculty, Federal University of Pará (UFPA), Belém. He coordinates the Research Laboratory in Computer Network and Multimedia Communication (GERCOM) from which he participates and coordinates several national and international projects. In 2020, he was a Visiting Professor with the University of Massachusetts at Amherst, Amherst, MA, USA, where he carried out his research on quantum networks. His research is focused on the field of networking. He is an Associate and a member of the Board of the Brazilian Computer Society (SBC).

Cristiano Bonato Both is a Professor of the Applied Computing Graduate Program with the University of Vale do Rio dos Sinos (UNISINOS), Brazil. He leads research projects funded by EU-Brazil, CNPq, FAPESP, MCTIC, and RNP. His research focuses on wireless networks, next-generation networks, softwarization, and virtualization technologies for telecommunication networks.

Flávio de Oliveira Silva (Member, IEEE) received the Ph.D. degree from the University of São Paulo in 2013. He is a Professor with the Faculty of Computing, Federal University of Uberlândia. He has had several papers published and presented at conferences worldwide. His main research interests are future networks, IoT, network softwarization (SDN and NFV), future intelligent applications and systems, cloud computing, and software-based innovation. He is a reviewer for several journals and a member of TCPs at several IEEE conferences. He is a member of ACM and SBC.

José A. Suruagy (Member, IEEE) graduated in electrical engineering from the Federal University of Pernambuco (UFPE) in 1979. He received the master's degree in electrical engineering from the University of São Paulo in 1982, and the Ph.D. degree in computer science from the University of California at Los Angeles in 1990. He is currently a Full Professor and a Researcher with the Centro de Informática—UFPE. He has experience in computer science, focusing on telecomputing, acting on the following subjects: computer networks, measurement, quality of service, and performance evaluation. In 2014, he received the "Computer Network Brazilian Symposium Distinguished Premium" for his scientific contributions in the areas of computer networks and distributed systems, for his participation in the symposium activities, and for his services that benefitted that community.

Rafael Pasquini (Senior Member, IEEE) received the M.Sc. and Ph.D. degrees in computer engineering from the State University of Campinas in 2006 and 2011, respectively. From 2015 to 2017, he was a Visiting Researcher with the Department of Network and Systems Engineering, KTH Royal Institute of Technology. Since 2011, he has been an Associate Professor and leads the Distributed Systems and Networks Research Group with the Department of Computer Science, Federal University of Uberlândia. His research interests include network management, slicing of softwarized infrastructures, machine learning, cloud computing, and software-defined networks.

Rodrigo Moreira received the B.S. degree from the Federal University of Viçosa in 2014, and the M.S. degree and the Ph.D. degree in computer science from the Federal University of Uberlândia, Brazil, in 2017 and 2021, respectively. He is a Professor with the Federal University of Viçosa (Campus Rio Paranaíba). His research interests include future Internet, quality of service, cloud computing, network function virtualization, software-defined networking, and edge computing.

Kleber V. Cardoso received the degree in computer science from the Universidade Federal de Goiás in 1997, and the M.Sc. and Ph.D. degrees in electrical engineering from COPPE—Universidade Federal do Rio de Janeiro in 2002 and 2009, respectively. He is an Associate Professor with the Institute of Informatics, Universidade Federal de Goiás, where he has been a Professor and a Researcher since 2009. In 2015, he spent his sabbatical with Virginia Tech, USA, and, in 2020, with Inria Saclay Research Center, France. His research interests include wireless networks, software-defined networking, virtualization, resource allocation, and performance evaluation.

Tereza C. Carvalho is an Associate Professor of Escola Politécnica University of São Paulo (USP) and a Visiting Professor with Université Paris 1 Panthéon-Sorbonne. She has been the Founder and the General Coordinator of LASSU (Laboratory of Sustainability on ITC) since 2010 and CEDIR-USP (Center for Reuse and Discard of Informatics Residuals) since 2009. She was a Former Assessor of CTI-USP (Information Technology Coordination) from 2010 to 2013 and the Director of CCE-USP (Electronic Computing Center) from 2006 to 2010. She is a Sloan Fellow 2002 from the Massachusetts Institute of Technology. She has coordinated international and national research and development projects since 2000 in Green Computing, Cloud Computing, IT Energy Efficiency, IT Governance, Digital Technologies applied to the Amazon Production Chains, Waste Electrical and Electronic Equipment (WEEE), Future Internet, Scientific DMZ, and Security. She holds several international patents.