Network Slicing Meets Artificial Intelligence: an AI-based Framework for Slice Management

Dario Bega, Marco Gramaglia, Andres Garcia-Saavedra, Marco Fiore,
Albert Banchs, and Xavier Costa-Perez

Abstract—Network slicing is an emerging paradigm in mobile networks that leverages Network Function Virtualization (NFV) to enable the instantiation of multiple virtual networks –named *slices*– over the same physical network infrastructure. The operator can allocate to each slice dedicated resources and customized functions that allow meeting the highly heterogeneous and stringent requirements of modern mobile services. Managing functions and resources under network slicing is a challenging task that requires making efficient decisions at all network levels, in some cases even in real-time, which can be achieved by integrating artificial intelligence (AI) in the network. We outline a general framework for AI-based network slice management, introducing AI in the different phases of the slice lifecycle, from admission control to dynamic resource allocation in the network core and at the radio access. A sensible use of AI for network slicing results in strong benefits for the operator, with expected performance gains between 25% and 80% in representative case studies.

1 Introduction

There is consensus among industry and standardization communities that network slicing [1] will represent a key paradigm in 5G mobile systems. Slicing allows the physical infrastructure to be "sliced" [2] into logical network instances, which are operated by different entities and may be tailored to support specific Quality of Service (QoS) requirements. A network slice is thus a collection of resources and functions that are orchestrated to support a specific service [3], encompassing software modules running in virtual machines, computational resources, and communication capacity in the backhaul and radio networks. Such modules and resources are customized to provide what is necessary for the service, avoiding unnecessary overheads and complexity.

The implementation of network slicing poses significant challenges from a technology viewpoint. The network infrastructure needs to be efficiently shared among different slices [4], while isolating slices from each other and customizing their functions to different requirements. This affects the operation of all of the different functions in the protocol stack, hence mandating that the design of mobile networks is completely re-visited.

Much of the complexity in re-designing mobile networks for slicing relates to decision-making towards an efficient, dynamic management of resources in real-time. As a matter of fact, sliced networks set out that a large number of logical network instances, each independently operated by a different tenant, must coexist within the same infrastructure and dynamically share the available physical resources. This drastically increases the complexity of the network management process with respect to legacy non-sliced systems controlled by a single entity, *i.e.*, the network operator, and renders traditional human-driven approaches inadequate. Instead, what operators need is an automated management of slices that: (i) takes advantage of the large volume of data flowing through the network and carrying information potentially relevant to a knowledgeable resource allocation; (ii) is proactive, by forecasting and exploiting the upcoming behavior of a system involving many different players.

In this paper, we discuss the potential utility of artificial intelligence (AI) for the management of sliced network, and propose a framework that integrates AI into different key functions of the system, described in Section 2. The proposed framework brings together three different AI-based solutions for network slicing, presented in Section 3 along with their possible shortcomings and countermeasures. The experimental results, discussed in Section 4, confirm that AI offers very effective and scalable solutions in multiple case studies of network slicing management. Our conclusion, drawn in Section 5, is that AI is well positioned to become a cost-effective approach in the context of sliced mobile networks, although there is an important margin of improvement in terms of trustworthiness and interpretability.

2 AI-based Slice Management Framework

The problem of managing a network slicing infrastructure can be decomposed into several fundamental tasks (Section 2.1), each of which can then be realized with suitable AI tools (Section 2.2).

2.1 Management functions in network slicing

Resource sharing across slices applies throughout all phases of the Network Slice Lifecycle Management [5], which consists of four main steps: (i) preparation; (ii) instantiation, configuration and activation; (iii) run-time; and, (iv) decommissioning. Our proposed framework covers three key functions that underlie the first three phases (the last phase does not involve management decisions):

1) *Admission control* is the critical decision-making mechanism during slice preparation, as it determines whether upcoming network slice requests can be admitted or not in the system, and is enacted so as to ensure that the requirements of the admitted slices are satisfied.

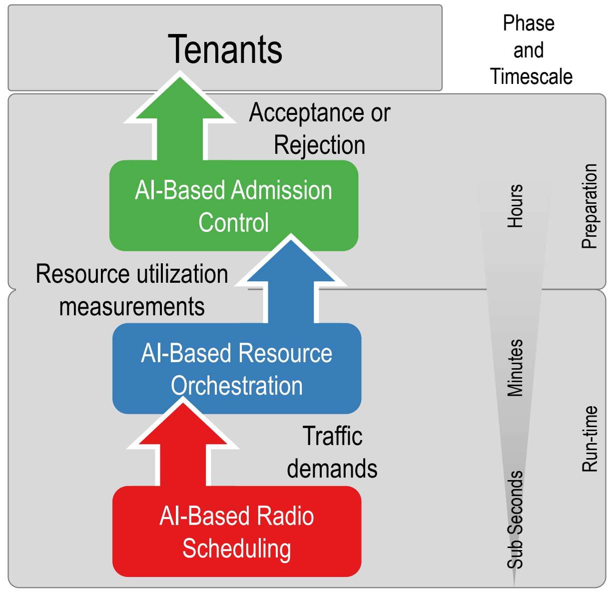

Fig. 1. Comprehensive network slicing framework. The diagram outlines the timescales and composition of the key slice management functions.

2) *Network resource (re)-orchestration* is central to both slice instantiation and run-time operation, since it allocates the available network resources to the admitted slices in the most efficient way possible, and then dynamically updates such an allocation at run-time in order to fulfill the time-varying demands of each slice while avoiding capacity outages.
3) *Radio resource scheduling* is paramount at run-time, as it manages the sharing of the radio access resources among the network slices, ensuring that the potentially stringent requirements of all the slices (*e.g.*, in terms of latency and throughput) are met over the air interface.

Figure 1 illustrates the functions above, each of which involves different timescales: (i) admission control runs at frequencies that match those of arrivals of new network slices requests, which may be in the order of hours; (ii) the orchestration of resources in softwarized networks occurs at frequencies that depend on the time required to re-sizing virtual machines resources, typically in the order of minutes; and, (iii) scheduling of radio resources applies at a finer granularity, down to millisecond intervals in extreme cases. We remark that different functions may interact among them. As an example, information on radio resource utilization can be leveraged for admission control, to understand if accepting new slices may cause shortages at the radio access.

2.2 Bringing AI into the picture

All functions above need to make decisions to meet the requirements of the individual slices while maximizing the overall system performance. To this end, they need to learn the dynamics of per-slice data traffic, and automatically react to their impact on the network architecture, towards their respective management goals. Self-adapting network function configurations were introduced over a decade ago, however the solutions designed so far typically apply control on limited sets of parameters that change slowly in time (*e.g.*, eNB transmission power). Also, current approaches produce outputs that then need human intervention to be translated into modifications of the network configuration (*e.g.*, updating the transport network so as to optimize handovers in a given region).

These characteristics are not compatible with the novel requirements introduced by network slicing. The parameters that may need reconfiguration are much more numerous, as each virtual network functions may expose several of them in a programmatic way. The timescale at which decisions must be made is drastically reduced, as one must be ideally capable of acting at radio level timings or even at wire-speed. Decisions often need to take into account metrics that go beyond pure network performance, such as energy efficiency or infrastructure monetization, which may hide complex cross-relationships.

This context provides a fertile ground for AI to become instrumental in mobile network operation. All classes of AI may be useful to this end, including (i) *supervised* solutions that require ground truth data for training, (ii) *unsupervised* techniques that work in absence of ground truth, and (iii) *reinforcement* learning approaches where different forms of interaction with the system that has to be controlled are possible [6]. The most appropriate AI tools must be selected case by case, depending on the involved algorithmic requirements and operation timescales.

For instance, reinforcement learning (RL) is particularly well suited when the time dynamics of the problem can accommodate a learning curve, and the objective is to define a sequence of actions that maximize a certain reward: this is the case in both admission control algorithm and radio resource scheduling, as demonstrated by the practical implementations presented in Sections 3.1 and 3.3, respectively.

Note that RL techniques need to be trained over a very large amount of data, which makes their application complex in real-world environments where historical data is scarce. Moreover, RL approaches suffer from the *curse of dimensionality*, when the underlying model becomes too large. Thus, in some cases, such as in the one detailed in Section 3.3, the usage of unsupervised learning combined with reinforcement learning may be needed to model very complex relationships in the input data. Conversely, when the target is to provide decisions that are independent of those previously taken and whose quality can be assessed during systems training, supervised learning solutions are a strong option: this is precisely the settings where network resource orchestration takes place, as illustrated by the applied solution in Section 3.2. Here, the challenge is to provide labelled data for the algorithm, which may be an unfeasible task if such labels cannot be directly obtained from the network.

Before proceeding further, we remark that those presented next are examples of successful integration of AI across the framework in Figure 1. They do not exhaust the application space of AI for network operations; rather, they realize important components in the comprehensive design of self-organizing sliced mobile networks.

3 AI-BASED SLICE MANAGEMENT FUNCTIONS

We present viable implementations of the slice management functions in our framework that rely on AI to perform admission control (Section 3.1), orchestration (Section 3.2), and radio resource scheduling (Section 3.3), respectively.

3.1 AI for admission control of slices

Network infrastructure resources are limited and network slice demand quality guarantees, which calls for admission control on new slice requests. According to 3GPP standardization on network slicing, the Communication Service Client (CSC) [5], *i.e.*, the tenant, will request specific services to the Communication Service Provider (CSP), *i.e.*, the network provider, among those available in the offered portfolio. Then, it will pay for the service according to metrics like, *e.g.*, the number of served users, the service coverage area, or the duration of the slice instance. Such admission control decisions have profound business implications: the choice of how many network slices to run simultaneously, and how to share the network infrastructure among slices have an impact on the revenues of the network provider.

During the admission control phase, a trade-off between resource sharing and KPI fulfillment needs to be tackled. If resource sharing is too aggressive, the required KPIs cannot be met and revenues drop as network slices do not provide the expected service; if instead network operators have exceedingly conservative approaches, they may miss substantial opportunities for profit.

Ultimately, the fact that stronger guarantees on KPIs require isolated, non-shared resources, draws a boundary on admissible configurations. In operational settings, this already tangled trade-off is further complicated by many technological and possibly time-varying variables, which makes finding the equilibrium point that maximizes revenues based on the admitted slices a difficult task. To identify the best operating point, slice admission control must learn the arrival dynamics of slices and make revenue-maximizing decisions based on the current system occupation and its expected long-term evolution.

Exact methods require that all the variables are known, and do not scale to large space states (which grow exponentially with the number of slices). Instead, AI provides apt tools to find the ridge between the maximum revenue per unit of time and the parameter space where KPIs are not met anymore, while proactively taking into account the whole set of time-varying relevant variables [7]. Specifically, Deep reinforcement learning (DRL) approaches interact with the large set of variables that characterize the system, infer its stochastic behavior, and then determine the best admission strategy according to the given target. A practical implementation is depicted in Figure 2. It leverages two fully-connected neural networks to estimate the average long-term rewards when admitting and rejecting a request, respectively [8]. More precisely, *feed-forward* neural networks (NNs) are used in both components, as a suitable architecture when dealing with function approximation. In a nutshell, the DRL solution operates as follows.

- Upon arrival of a new slice request, the system takes the action (*i.e.*, accepting or rejecting the request) that maximizes long-term revenue; each NN is in charge of forecasting the revenue associated to one action.
- After an action is taken, the algorithm interacts with the system and evaluates the quality of the generated revenue through a specific loss function. This value is then utilized to train the corresponding neural network.
- During such a training phase, the *wrong* action (*i.e.*, the one related with a lower long-term revenue) is used instead of that actually performed with a low probability ϵ, which allows exploring the system and adjusts to its variations over time.
- When a substantial change in the system behavior is observed, a complete re-training of the system may be triggered to adapt the algorithm to the new conditions.

This AI-based approach has a number of advantages, including: (i) flexibility in adapting to system settings changes thanks to short convergence time; (ii) effective operation in situations that were never met before, thanks to the reliable estimations provided by the feed-forward NN architectures in those cases; (iii) scalability to large network sizes.

3.2 AI for network resource orchestration

Once admitted, slices must be allocated sufficient resources. Due to the prevailing softwarization of mobile networks, such resources are increasingly of computational nature. Ensuring strong KPI guarantees often requires that computational resources are exclusively allocated to specific slices, and cannot be shared across others [9]. The dynamic allocation of network resources to the different admitted slices is, in fact, a chief management task in network slicing.

In this context, the network operator needs to decide in advance the amount of resources that should be dedicated to the different slices, so as to ensure that the available capacity is used in the most efficient way possible and thus minimize operating expenses (OPEX). The key trade-off is between:

- *underprovisioning* – if the operator allocates less capacity than that required to accommodate the demand, it incurs into violations of the Service Level Agreement (SLA) established with the tenant;
- *overdimensioning* – excess resources assigned to a slice imply a cost in terms of unnecessarily allocated resources that go unused.

Finding the correct operational point requires (i) predicting the future demand in each slice [10], and (ii) deciding what amount of resources is needed to serve such demand. These two problems are complex per-se: forecasting future demands at service level requires designing dedicated, accurate predictors; instead, allocating resources in a way that underprovisioning and overdimensioning are poised to minimize the OPEX of the operator requires estimating the expected (negative and positive) error of the prediction. Moreover, addressing (i) and (ii) above as separate problems, risks to lead to largely suboptimal solutions, since legacy predictors do not provide reliable information about the expected error they will incur into.

While the complexity of the complete solution may be daunting with traditional techniques, AI can be leveraged to address both aspects at once, by solving a *capacity forecast* problem. This can be realized by training a typical Convolutional Neural Network (CNN) architecture for time series prediction with a dedicated loss function that, instead of simply minimizing the error, accounts for the respective costs of SLA violations and capacity overprovisioning [11]. Note that we employ a CNN-based architecture instead

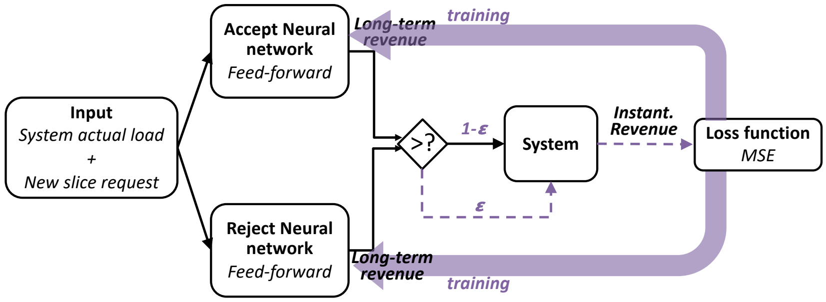

Fig. 2. High-level design of AI-based slice admission control.

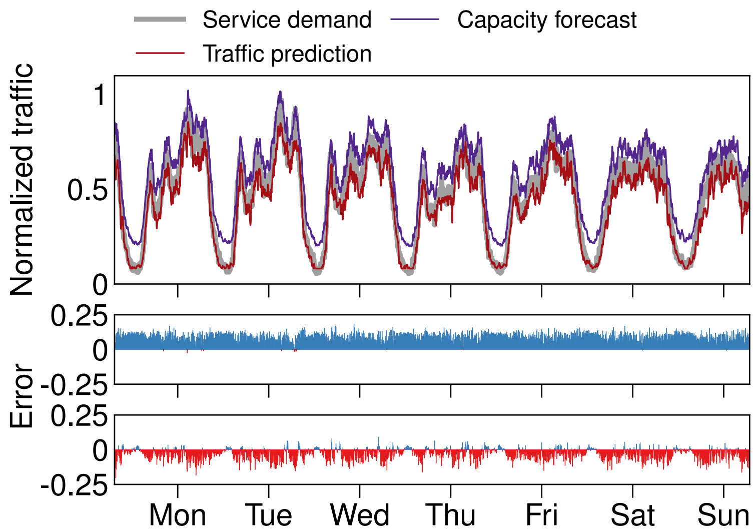

Fig. 3. Top: Predictions of a sample one-week demand, as produced by a legacy MAE traffic predictor and by a capacity forecasting model. Middle: Error incurred by the capacity forecasting model, which only generates overprovisioning. Bottom: Error incurred by the MAE traffic predictor, which leads to frequent service requirement violations.

of Recurrent Neural Network (RNN) models traditionally used for regression problems. Unlike RNN, CNN allows exploiting inherent spatial correlations in the traffic generated at different geographical locations. Moreover, we adopt a 3D-CNN architecture that can accommodate a tensor input; by having time as the third dimension, the model can properly account for relevant temporal autocorrelations.

The operation of the proposed approach is exemplified in Figure 3, where a typical traffic prediction minimizing the Mean Absolute Error (MAE) is compared with a capacity forecast that accounts for the actual costs of underprovisioning and overdimensioning. As shown in the top plot of Figure 3, the network trained with MAE tries to anticipate the exact demand. By doing so, it incurs in expensive underprovisioning during a substantial portion of the time, as depicted in the bottom plot. Instead, the network trained with Deepcog can learn how to dimension capacity in the next time slot so as to avoid SLA violations, while keeping overprovisioning at a minimum, as also illustrated in the middle plot. Such a properly tuned AI-based solution allows determining the resources that should be proactively allocated to each slice to accommodate their future demand. Ultimately, this approach solves (i) and (ii) at once via AI, while minimizing the overprovisioning, and avoiding the service requirement violations generated by a legacy traffic predictor, with major monetary savings for the operator.

3.3 AI for slice scheduling at radio access

At radio access, a key challenge of network slicing is the design of a RAN virtualization (vRAN) mechanism that jointly (i) provides isolation between network slices, and (ii) adapts the allocation of pooled physical resources to the needs of each virtual RAN. As highlighted in Section 2, this must occur at much faster timescales than those considered before. Thus, the design of algorithms that implement adaptive policies to multiplex resources is paramount.

However, optimizing the allocation of resources is particularly challenging as there is a strong non-linear coupling between computing and radio control policies, which makes it very hard to determine the compute resources that should be allocated to a slice depending on the amount of radio resources dedicated to it. The complicated relationship between radio and compute resources is made evident by the experimental results shown in Figure 4. The plots depict the uplink throughput performance in high-SNR regimes of a virtualized LTE BBU for a wide set of radio (abscissa) and computing (ordinate) allocation policies and different data load conditions. The plots refer to two different compute nodes hosting the BBU. These observations imply that traditional modeling approaches, which require pre-calibration for specific conditions and platforms, are not appropriate for practical (v)RAN slicing. This is because the coupling between Modulation and coding schemes (MCS) and CPU

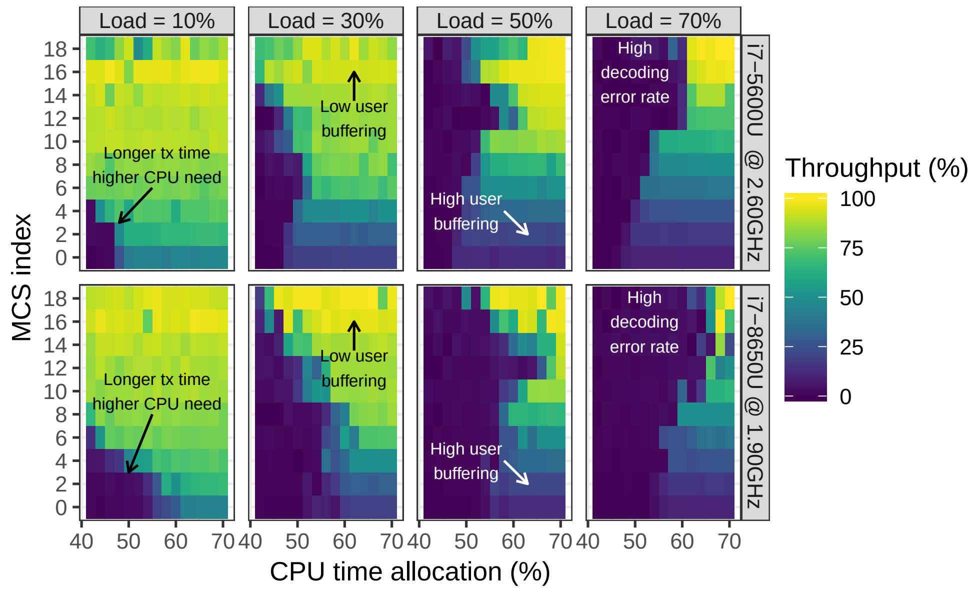

Fig. 4. Virtualized LTE BBU with high SNR. Performance model is highly complex and non-linear. Reproduced from [12].

time is far from trivial and depends on SNR conditions, traffic load and the platform hosting the BBU.

In this context, a combination of unsupervised learning and deep reinforcement learning is a promising solution [12]. Indeed, unsupervised learning methods such as deep auto-encoders (and popular variants) are particularly helpful to project high-dimensional contextual data such as data load patterns or SNR patterns into (sparse) latent spaces, while deep reinforcement learning algorithms design policies that map such (encoded) context information into optimal resource control actions like, *e.g.*, CPU or radio scheduling decisions, which handle the aforementioned challenges. Building on these techniques, a deep deterministic policy gradient (DDPG) algorithm, implemented by actor-critic neural network structures, can deal with large and/or continuous action spaces, which are common in resource control problems [13]. More precisely, actor-critic algorithms are the most suitable type of solution for the class of problems we are addressing as they combine the advantages of both RL classic approaches (policy based), which support large action spaces, and value-based RL, which provides higher sample-efficiency and stability. Actor-critics split the model into (i) the actor that takes the context as input and learns the optimal policy and (ii) the critic evaluates the action by estimating the value function.

Resource control can be indeed formulated as a contextual bandit problem, which is a particular case of RL. There, one observes a *context* vector, chooses an action and receives a reward signal as feedback, sequentially at different time stages. The goal is to find a policy that maximizes the expected reward. In our case we adopt the following formulation for the context, action and reward [12].

- *State or context space.* At each stage, several samples, representatives of the current context (*i.e.*, the channel conditions and the load) are collected.
- *Action space.* Figure 4 (and additional results in [12]) illustrates the strong coupling between CPU control decisions (x-axis in Figure 4) and the MCS used for radio communication (y-axis in Figure 4). As a result, two sets of actions are designed, (i) *Computing control decision*, which allocate the necessary CPU to support the (ii) *Radio Control decisions*, which fix an upper-bound eligible MCS index. The action space comprises all pairs of compute and radio control decisions.
- *Rewards.* We designed the reward function to weight and balance the costs related to higher delays, decoding errors and CPU utilization with the coal of (i) minimizing operational costs due to CPU reservations, and (ii) maximizing performance by reducing decoding error rates and latency.

Figure 5 illustrates the decision-making closed-loop process implementing the RL formulation above.

4 Trial Gains of AI-based Slice Management

A frequent controversy of integrating AI in network systems is whether these novel architectures actually bring a substantial advantage in performance over traditional approaches, sufficient to justify the added complexity and loss of interpretability. Network slicing is no exception, hence we conclude our discussion with quantitative figures that are representative of the gain margin that AI can provide when applied to the slice management functions presented in Sections 3.1–3.3.

4.1 Increasing the infrastructure monetization with optimal admission policies

We first evaluate the advantages brought by the intelligent admission control of network slices from a monetary perspective. Specifically, we analyze the revenues obtained by a provider when facing the decision of accepting or not two kinds on network slices: one (eMBB, less expensive) including elastic traffic and one (URLLC, more expensive) including inelastic traffic with strong QoS requirements. Results considering admission requests that follow a Poisson process for both kinds of slices are summarized in Table 1.

We compare the AI-based algorithm described in Section 3.1 with two benchmarks: (i) the optimal admission

TABLE 1
Summary of gains attained by AI over legacy solutions for sliced network management. Full results are available in [8], [11], [12].

Function	Performance metric	Percent improvement – Benchmark use case
Network Slice Admission Control	Revenue improvement	$-\mathbf{0.23}\%$ – Optimal, ratio 1 $-\mathbf{3.77}\%$ – Optimal, ratio 20 $\mathbf{33.3}\%$ – Random Policies, ratio 15
Cloud resource allocation	Reduction of monetary cost	$\mathbf{81.6}\%$ – Facebook, Core datacenter $\mathbf{59.2}\%$ – Snapchat, MEC datacenter $\mathbf{64.3}\%$ – YouTube, C-RAN datacenter
Virtual RAN resource allocation	CPU savings, delay, throughput	$\mathbf{30}\%$ – CPU savings over CPU-blind schedulers $\mathbf{25}\%$ – Delay-based QoS over CPU-blind schedulers $\mathbf{25}\%$ – Throughput upon computational capacity deficit

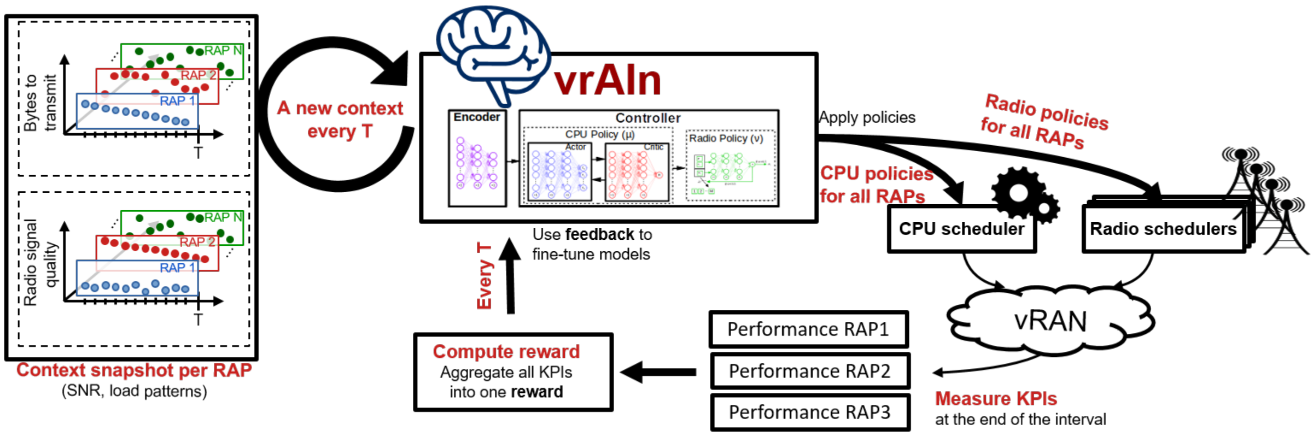

Fig. 5. AI-based closed control-loop vRAN resources management for slicing [12].

policy, computed through the expensive value-iteration algorithm, and a (ii) set of smart random policies (which mimic a large number of heuristics, 10,000 in our setup). Experiments are performed for different price ratios between the URLLC and eMBB (where a ratio x means that URLLC are x times more expensive than eMBB) and different network sizes (the comparison against the optimal policy is performed on a smaller scale scenario as the optimal policy cannot be run for large scenarios). Results show that the AI-based algorithm described in Section 3.1 takes intelligent admission decisions and provides (i) very substantial gains over heuristic approaches, (ii) while incurring into an almost negligible penalty when compared to an optimal, not feasible in practice, benchmark.

4.2 Real world and data-driven capacity forecasting

To assess the performance of AI for the orchestration of sliced network resources described in Section 3.2 , we consider three representative case studies: (i) a slice dedicated to Facebook traffic in a core network datacenter that controls all 470 4G eNodeBs deployed in a large metropolitan area; (ii) a slice allocated to Snapchat traffic in a Mobile Edge Computing (MEC) datacenter that handles the traffic of around 70 eNodeBs; and (iii) a slice accommodating traffic generated by the YouTube app at a C-RAN datacenter located in proximity of the radio access, which performs baseband processing and scheduling for 11 eNodeBs. We use as a benchmark a legacy mobile traffic predictor, which minimizes the MAE of the forecast. The prediction returned by this model is overprovisioned by 5% –a reasonable figure in presence of decently accurate prediction– to try avoiding SLA violations. The results are summarized in Table 1.

The AI-based approach to network resource orchestration outlined in Section 3.2 achieves a substantial reduction of the monetary cost associated to the allocation of resources, with savings above 50% in all cases.

4.3 AI-based CPU optimization at the edge

Finally, the joint radio/CPU scheduling approach described in Section 3.3 is evaluated by means of an experimental proof-of-concept. The testbed comprises a set of software-defined radio (SDR) USRPs as radio front-ends connected to a computing node hosting a softwarized LTE eNodeB BBU, namely, `srsLTE eNB` [14]. We selected this software platform mostly for its flexiblity in the internal software design [15]. The aforementioned DDPG reinforcement learning algorithm and interface with the eNodeB are implemented through (i) Docker's API to control the allocation of computational resources of each stack (specifically, to govern the CFS CPU scheduler used by Docker), and (ii) a simple socket to induce bounds on the eligible MCS by the UEs associated to the eNB.

Two scenarios are considered for performance evaluation. In the first scenario, computational capacity is unbounded and the goal is to strive a good balance between system cost (CPU usage) and QoS performance (a proxy for the system delay). We can observe that, even in highly dynamic scenarios, an AI-based approach can achieve a 25% improvement on QoS performance over static CPU allocation policies employing the same amount of computational resources in average, and up to 30% of CPU average savings with minimal QoS penalty over CPU-blind schedulers. In the second scenario, we constrain the computational resources running available to two competing eNBs. In this setup, AI ensures 25% more throughput with almost zero decoding errors.

5 Conclusions

In this paper, we discussed the potentially critical role of AI for the management and operation of mobile networks that implement network slicing. AI-based solutions can address the different and very complex problems that emerge at multiple levels, including scheduling of slice traffic at RAN, resource allocation to slices in the network core, and admission control of new slices. We outlined practical deep learning architectures that can solve such problems in three different case studies, and illustrated the high typical gain that one can expect from integrating AI in network slicing. We conclude that AI has a clear potential to become a cardinal technology for future-generation zero-touch mobile networks. This also implies that present limitations of AI architectures in general, *e.g.*, curbed trustworthiness and interpretability, shall be seriously considered and solved, ensuring that operators retain full control over their systems.

Acknowledgments

The work of University Carlos III of Madrid was supported by H2020 5G-TOURS project (grant agreement no. 856950). The work of NEC Laboratories Europe was supported by H2020 5GROWTH project (grant agreement no. 856709). The research of Marco Fiore was partially supported by the ANR CANCAN project (ANR-18-CE25-0011).

References

[1] D. Bega, M. Gramaglia, C. J. Bernardos Cano, A. Banchs, and X. Costa-Perez, "Toward the network of the future: From enabling technologies to 5G concepts," *Transactions on Emerging Telecommunications Technologies*, vol. 28, no. 8, p. e3205, Aug. 2017.

[2] K. Samdanis, X. Costa-Perez, and V. Sciancalepore, "From network sharing to multi-tenancy: The 5G network slice broker," *IEEE Communications Magazine*, vol. 54, no. 7, pp. 32–39, 2016.

[3] V. Sciancalepore, K. Samdanis, X. Costa-Perez, D. Bega, M. Gramaglia, and A. Banchs, "Mobile traffic forecasting for maximizing 5G network slicing resource utilization," in *Proc. of IEEE INFOCOM*, May 2017, pp. 1–9.

[4] C. Marquez, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "How Should I Slice My Network?: A Multi-Service Empirical Evaluation of Resource Sharing Efficiency," in *Proc. of ACM MobiCom*, Oct. 2018, pp. 191–206.

[5] 3GPP, "Telecommunication management; Study on management and orchestration of network slicing for next generation network," 3GPP, Technical Report (TR) 28.801, 2018. [Online]. Available: http://www.3gpp.org/DynaReport/28801.htm

[6] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016, http://www.deeplearningbook.org.

[7] J. X. Salvat, L. Zanzi, A. Garcia-Saavedra, V. Sciancalepore, and X. Costa-Perez, "Overbooking Network Slices Through Yield-driven End-to-end Orchestration," in *Proc. of ACM CoNEXT*, Dec. 2018, pp. 353–365.

[8] D. Bega, M. Gramaglia, A. Banchs, V. Sciancalepore, and X. Costa-Prez, "A Machine Learning approach to 5G Infrastructure Market optimization," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2019.

[9] J. Mendes, X. Jiao, A. Garcia-Saavedra, F. Huici, and I. Moerman, "Cellular access multi-tenancy through small-cell virtualization and common rf front-end sharing," *Computer Communications*, vol. 133, pp. 59–66, 2019.

[10] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications Surveys & Tutorials*, Mar. 2019.

[11] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez, "DeepCog: Cognitive Network Management in Sliced 5G Networks with Deep Learning," in *Proc. of IEEE INFOCOM*, May 2019.

[12] J. A. Ayala-Romero, A. Garcia-Saavedra, M. Gramaglia, X. Costa-Perez, A. Banchs, and J. J. Alcaraz, "vrAIn: A Deep Learning Approach Tailoring Computing and Radio Resources in Virtualized RANs," in *Proc. of ACM MobiCom*, Oct. 2019.

[13] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[14] I. Gomez-Miguelez, A. Garcia-Saavedra, P. D. Sutton, P. Serrano, C. Cano, and D. J. Leith, "srsLTE: An Open-source Platform for LTE Evolution and Experimentation," in *Proc of ACM WiNTECH*, Oct. 2016, pp. 25–32.

[15] F. Gringoli, P. Patras, C. Donato, P. Serrano, and Y. Grunenberger, "Performance Assessment of Open Software Platforms for 5G Prototyping," *IEEE Wireless Communications*, vol. 25, no. 5, pp. 10–15, 2018.

Dario Bega holds a Ph.D. from IMDEA Networks Institute and University Carlos III of Madrid (2020).He received his M.Sc (2013) degrees in Telecommunication Engineering from the University of Pisa, Italy.

Marco Gramaglia is a post-doc researcher at University Carlos III of Madrid, where he received M.Sc (2009) and Ph.D (2012) degrees in Telematics Engineering.

Andres Garcia-Saavedra received his M.Sc and Ph.D. from University Carlos III of Madrid in 2010 and 2013, respectively. Since July 2015 he is a Senior Researcher at NEC Laboratories Europe.

Marco Fiore (S'05, M'09, SM'17) is a Reseach Associate Professor at IMDEA Networks Institute (Spain). He received a PhD from Politecnico di Torino (Italy), and a HDR from Univeristé de Lyon (France). He held tenured positions at INSA Lyon (France) and CNR (Italy).

Albert Banchs (M'04-SM'12) received the M.Sc. and Ph.D. degrees from the Polytechnic University of Catalonia. He is Full Professor with the University Carlos III of Madrid, and Deputy Director of the IMDEA Networks institute.

Xavier Costa-Pérez (M'01) is Head of 5G Networks R&D at NEC Laboratories Europe.He received both his M.Sc. and Ph.D. degrees in Telecommunications from the Polytechnic University of Catalonia.