AI-Based Autonomous Control, Management, and Orchestration in 5G: From Standards to Algorithms

Dario Bega, Marco Gramaglia, Ramon Perez, Marco Fiore, Albert Banchs, and Xavier Costa-Pérez

Abstract

While the application of artificial intelligence (AI) to 5G networks has raised strong interest, standard solutions to bring AI into 5G systems are still in their infancy and have a long way to go before they can be used to build an operational system. In this article, we contribute to bridging the gap between standards and a working solution by defining a framework that brings together the relevant standard specifications and complements them with additional building blocks. We populate this framework with concrete AI-based algorithms that serve different purposes toward developing a fully operational system. We evaluate the performance resulting from applying our framework to control, management, and orchestration functions, showing the benefits that AI can bring to 5G systems.

Introduction

Network control, management, and orchestration entail the dynamic placement, configuration, and resource provisioning of virtual network functions (VNFs) within the network function virtualization (NFV) infrastructure. The complexity of these operations exceeds substantially that of equivalent tasks in legacy 4G LTE networks. There, the relatively limited amount of variables in one-size-fits-all in the core and radio access network domains accommodates management models that mainly rely on expert monitoring and intervention. Instead, the traditional human-based approach is hardly viable in virtualized 5G networks where the coexistence of heterogeneous mobile services, diversified network requirements, and tenant-defined management policies create a need for specialized and time-varying infrastructure deployments. This calls, in turn, for automated solutions in the control, management, and orchestration of the network.

Artificial intelligence (AI) is a natural choice to support the emerging need for autonomous network operation and management. The Third Generation Partnership Project (3GPP) and other standards development organizations (SDOs) have started delineating the road for the integration of AI into the mobile network architecture. Such a process starts with an efficient collection of data in the network infrastructure and knowledge inference from these data, which are paramount to effective AI-assisted decision making. In this sense, SDOs are pushing efforts toward defining AI-based data analytics frameworks that are suitable for autonomous and efficient control, management, and orchestration of mobile networks. For instance, 3GPP has incorporated the following modules into its standardized architecture: (i) network data analytics function (NWDAF) [1] and (ii) management data analytics function (MDAF) [2]. Other organizations, such as the O-RAN alliance, envision similar entities in their architectures [3]. The European Telecommunications Standards Institute (ETSI) has also defined comparable assisting elements within the Industry Specification Groups (ISGs) on Experiential Networked Intelligence (ENI) and Zero touch network & Service Management (ZSM) [4]. Furthermore, open source initiatives such as ONAP [5] are also including data analytics in their architecture.

All these ongoing efforts are, however, at an early stage. The frameworks they propose and the solution designs they foster are preliminary and mainly aim at introducing several key building blocks at a very high level of abstraction. They are still far from detailed, full-blown network data analytics that are ready for deployment.

In this context, the goal of this article is to complement and support ongoing standardization activities by (i) proposing a comprehensive framework that leverages data analytics for network control, management, and orchestration, bringing together the corresponding efforts at relevant initiatives such as 3GPP, ETSI, and O-RAN; and (ii) populating the proposed framework with practical algorithms that build on AI and machine learning (ML) solutions.

AI-Driven Data Analytics Framework

Figure 1 depicts the network data analytics framework we propose. The framework design encompasses the management and orchestration plane as well as the control plane functionalities, as AI can indeed improve the performance at all levels. Within each plane, we take as reference architecture the one proposed by 3GPP, integrating it with an ETSI NFV MANO architecture and expanding it with O-RAN modules.

Management and Orchestration Plane

In the management and orchestration plane, the MDAF module is responsible for the so-called management data analytics service (MDAS) for all network slice instances, sub-instances, and net-

Digital Object Identifier:
10.1109/MNET.001.2000047

Dario Bega and Albert Banchs are with IMDEA Networks Institute and University Carlos III of Madrid; Marco Gramaglia is with University Carlos III of Madrid; Ramon Perez is with Telcaria Ideas; Marco Fiore is with with IMDEA Networks Institute; Xavier Costa-Pérez is with NEC Laboratories Europe, i2CAT, and ICREA.

0890-8044/20/$25.00 © 2020 IEEE

work functions hosted within the network infrastructure. This involves the centralized collection of network data for subsequent publishing to other network management and orchestration modules. In the proposed framework, we specifically employ this service to collect mobile data traffic loads generated in the radio access domain by the individual slices; in particular, the MDAS [2] comprises the load level at both the network function (NF) and network slice levels, provided as a periodic notification and expressed either in absolute terms or relative to the provisioned capacity. As a result, the MDAF allows building historical databases of the network demands for each base station and slice. These data are then exposed to the AI-based prediction algorithms for long-term forecasting (`AI-LTF`) and mid-term forecasting (`AI-MTF`).

The `AI-LTF` algorithm aims to assist the VNF placement decisions made by the orchestration system. To this end, `AI-LTF` leverages the network demand history to predict the future aggregate load across the different infrastructure locations. Then, the NFV orchestrator (NFVO) compares this prediction against the current available capacity in each infrastructure location and anticipates potential overload conditions. The NFVO can react, for example, by moving VNFs out of the congested infrastructure (while meeting the requirements of the corresponding network slice). The `AI-LTF` algorithm operates on long timescales, typically on the order of hours: indeed, VNF repositioning is quite a drastic action that involves substantial overhead, and consequently it is only performed infrequently and as an answer to substantial traffic fluctuations.

The second algorithm, `AI-MTF`, has a different purpose: it fuels the resource scaling decisions made by the VNF manager (VNFM). The VNFM has an interface with the virtual infrastructure managers (VIMs) to monitor the resource usage of the VNFs of each slice, and it also leverages data collected and published by the MDAF to determine the level of unsatisfied demand and the amount of unused resources. Based on all this information, the `AI-MTF` algorithm assists the orchestration framework on the decision to provide more resources to the VNFs of a slice when the predicted load exceeds the current resources, an operation typically referred to as *upscaling*, or to *downscale* resources to save cost when VNFs are leaving a significant fraction of the resources unused. Such decisions must be made over faster timescales than those affecting the VNF placement, and generally occur over intervals on the order of tens of minutes, which is the typical frequency for the execution of new VNF instances involving up- and downscaling.

Note that `AI-LTF` and `AI-MTF` only take as input the load history from MDAF and do not interact between themselves or with any other module. The forecasts of `AI-LTF` and `AI-MTF` are fed into the NFVO and VNFM engines, which may instead also leverage information obtained from other modules to make their decisions.

Control Plane

On the control plane, the NWDAF module is responsible for collecting data on the load level of an NF or a network slice [1], playing a very similar role to that of the MDAF in the management domain. In our framework, these data are fed to the AI-based short-term forecasting algorithm (`AI-STF`), which predicts the future traffic load. The forecast is leveraged by the Policy Control Function (PCF) module, which provides a unified policy framework to govern the network behavior. PCF can use the forecast provided by `AI-STF` to optimize its policies, such as:

- The QoS parameters (for those services that can be provided at different quality of service, QoS, levels)
- The access and mobility policies
- The user equipment route selection policy (URSP)

In contrast to the previous modules, these updates are performed at rather fast timescales, down to hundreds of milliseconds.

While the NWDAF module has been designed for the network core, a similar approach can be applied to the radio access network (RAN). Although 3GPP has not yet proposed modules equivalent to NWDAF in the RAN, other initiatives such as the O-RAN alliance have taken this path. In the O-RAN architecture [3], the radio network information base (RNIB) collects load information of flows or flow aggregates at the RAN level, the RAN intelligent controller (RIC) enables near-real-time control of RAN elements/resources, and the RAN resource orchestrator handles the overall resources at the base station level. In this case, the `AI-STF` forecasts can be leveraged by the RIC to perform the optimization of the radio resources at a fine time granularity (on the order of hundreds of milliseconds) and by the RAN resource orchestration to update the resource and bandwidth allocation at larger timescales (up to the order of minutes).

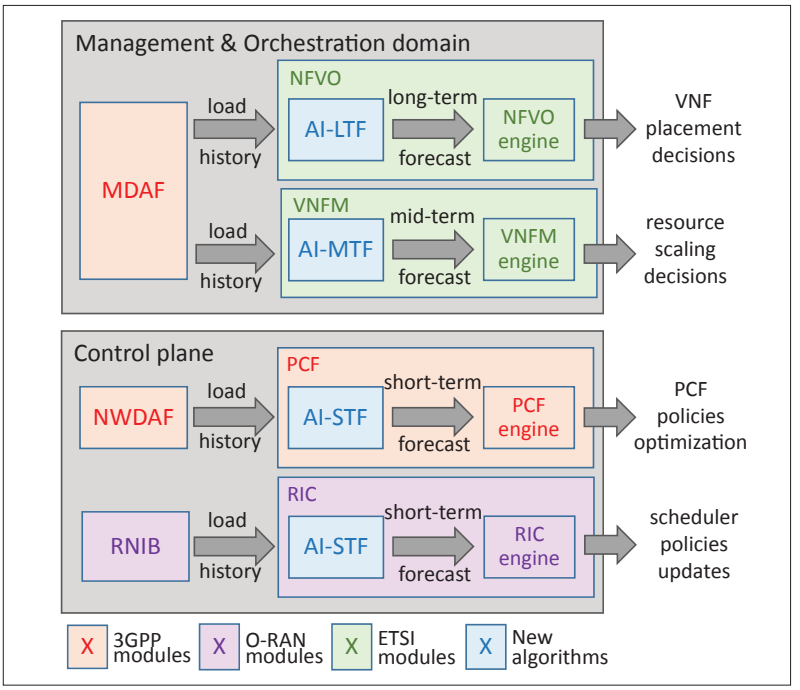

FIGURE 1. Proposed framework with standard functions (from 3GPP, ETSI, and O-RAN) and the new AI-based algorithms (`AI-LTF`, `AI-MTF`, and `AI-STF`).

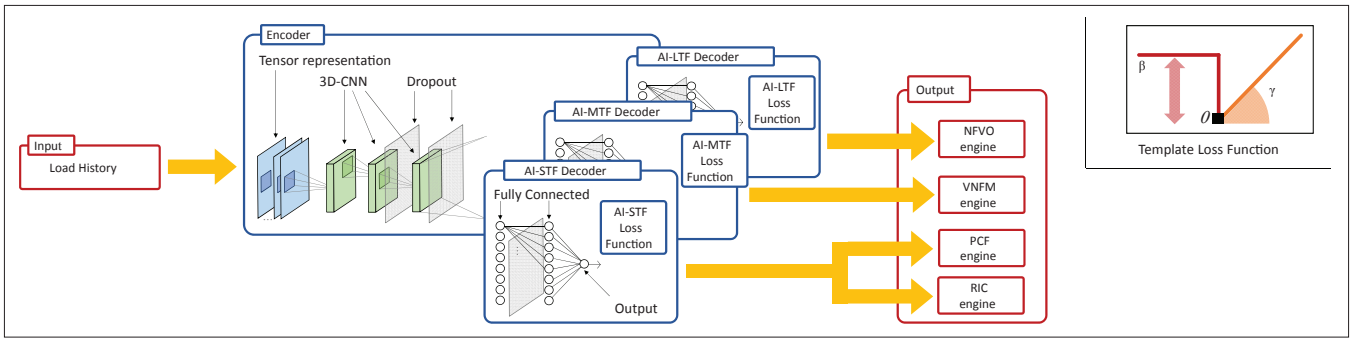

FIGURE 2. Neural network encoder-decoder structure.

AI-Based Algorithm Design

The above framework introduces three new AI-based algorithms: `AI-LTF`, `AI-MTF`, and `AI-STF`. The three algorithms follow the same design guidelines, as all of them aim to provide network capacity forecasts. The main difference between them is that they work at different granularity in terms of traffic volume (at global, slice, or flow levels) and timescale (intervals of hours, tens of minutes, minutes or shorter). In the following, we present the unified design of these three algorithms.

Capacity Forecasting

In contrast to the majority of the literature in the area of forecasting, our algorithm design addresses an original problem of capacity forecasting. Capacity forecasting goes beyond the typical estimation of future demands that is targeted by most traffic predictors. Indeed, predictors in the literature almost exclusively aim at matching the temporal behavior of traffic as closely as possible, giving the same weight to positive and negative errors [6]. While this approach produces forecasts that reduce as much as possible the error between the future and the anticipated demand, it is unsafe in a capacity allocation context where the metric of interest is the cost incurred by an operator when deploying the resources, rather than the error between the real and the forecasted demand. In this case, underestimating future demands causes service level agreement (SLA) violations that have a monetary penalty much higher than the cost resulting from overprovisioning the resources, as long as the level of overdimensioning is not excessive.

In contrast to the above legacy approaches, the aim of capacity forecasting is to find the level of capacity that suffices to meet the expected load at (almost) all times, even if this comes at the price of requiring a certain level of overprovisioning. To perform such capacity forecasting, we rely on AI techniques, which have been repeatedly shown to outperform traditional statistical models in mobile traffic prediction tasks that are akin to the capacity forecasting problem at hand [6, 7]. In particular, our design takes advantage of recent advances in supervised learning via deep neural network (DNN) architectures, which — unlike other approaches — are well suited to cope with the high dimensionality of the mobile data traffic, the complex spatial and temporal correlations it entails [8], and the nonlinear metric of interest that characterizes our problem.

Algorithm Design Overview

Our algorithm design builds on recent proposals that properly model the monetary costs incurred by the mobile network operator [9]. It is based on the following workflow. First, current and past mobile traffic information, collected at the desired level of granularity, is properly formatted into an *input* suitable for feeding the prediction algorithm. This input is fed to a *DNN architecture* that processes input features to provide an *output* value: the capacity forecast. During the training phase, the output is used to evaluate a *loss function* that quantifies the error with respect to the ground truth (i.e., the label), accounting for the costs of resource overprovisioning (i.e., allocating more capacity than needed) and underprovisioning (i.e., allotting insufficient capacity to meet the demand).

More precisely, time is divided into slots, and data on the actual traffic load is collected by MDAF, NWDAF, and RNIB for each slot. The collected load refers to the total load (for the `AI-LTF` algorithm), the load of individual slices (for the `AI-MTF` algorithm), and the load of flows or flow aggregates (for the `AI-STF` algorithm). Base stations are associated with data centers such that a data center serves the aggregated load of all its associated base stations. Our framework aims at allocating the required capacity at each data center or associated NFs.

Our goal is to compute a *constant* capacity to be allocated in the network data centers over a future time horizon T_h, based on knowledge of the previous T_p traffic snapshots. The time horizon models typical situations where the resource reconfiguration frequency is limited (e.g., by the NFV technology), and the operator must decide in advance the amount of resources that will stay assigned to a slice until the next reallocation takes place. As discussed before, `AI-STF`, `AI-MTF`, and `AI-LTF` target short, intermediate, and long time horizons, respectively.

To perform capacity forecasting, we leverage a DNN composed of suitably designed encoding and decoding phases, which operate over an interval T_h. The neural network architecture is general enough that it can be trained to solve the capacity forecast problem for traffic loads with diverse demand patterns, any data center, and any time horizon T_h. This allows leveraging the same DNN design to implement all three algorithms. The design consists of the following three components.

Encoder: The historical mobile data traffic provided as input is high-dimensional, as it comprises

a large number of base stations as well as several network slices or flows. The encoder projects this complex input space into a latent low-dimensional representation, which is then analyzed to produce the needed prediction.

Decoder: The decoder performs the actual forecast. The decoder structure reflects the kind of output values that shall be used to assist our framework, including the traffic granularity (i.e., the data center and the traffic volume level) and the time horizon.

Loss Function: The supervised learning strategy we adopt requires that the algorithm can assess the goodness of the outcome. To this end, we employ a dedicated loss function to measure the quality of the capacity forecast and steer the system during the training phase.

In the remainder of this section, we detail the implementation of the above three components. While the three algorithms considered in this article (`AI-LTF`, `AI-MTF`, and `AI-STF`) share the same encoder structure, they output the forecasts over different time horizons, which has an impact on the decoder and the loss function computation.

Encoder and Decoder Structure

The neural network architecture used by the proposed algorithms is summarized in Fig. 2, and is composed of an encoder-decoder sequence. The internal structures of the encoder and decoder are inspired by recent breakthroughs in deep learning for image and video processing [10]. Their design stems from the intuition that subsequent snapshots of the spatial distribution of the network data traffic can be assimilated to frames in a video.

The encoder is composed of a stack of three three-dimensional convolutional neural network (3D-CNN) layers [10]. CNNs are a kind of deep learning structure specialized to infer local patterns in the feature space of a matrix input. Two-dimensional CNNs (2D-CNNs) have been extensively utilized in image processing to complete complex tasks on pixel matrices such as face recognition and image quality assessment. 3D-CNNs extend 2D-CNNs to address the case where the features to be learned are spatiotemporal in nature, which adds the time dimension to the problem and transforms the input into a 3D-tensor.

Since mobile network traffic exhibits correlated patterns in space and time, we design an encoder that employs 3D-CNN layers. We use a $3 \times 3 \times 3$ kernel for the first 3D-CNN layer and a $6 \times 6 \times 6$ kernel for the second and third layers. This limits the portion of input analyzed by each neuron to small regions — a strategy known to perform well when the input has strong local correlations. We employ ReLU activation functions, which grant good performance and fast learning [11].

The decoder uses multi-layer perceptrons (MLPs) [12], a class of fully connected neural layers where every neuron of one layer is connected to every neuron of the next layer. MLPs are able to learn global patterns in the input feature space, which allows forecasting the target capacity leveraging the local features extracted by the encoder. For the decoder activation functions, we employ ReLU in all MLP layers except for the last one, where a linear activation function returns real-valued outputs. The last linear layer is capable of performing multiple capacity forecasts in parallel (e.g., for different slices or different data centers).

For the training procedure, we employ the popular Adam optimizer, a stochastic gradient descent (SGD) method with fast convergence properties [13]. This trains the neural network model by evaluating at each iteration the loss function resulting from the forecast and the ground truth, and back-propagating it to tune the model parameters to minimize such loss.

Loss Function Design

The loss function drives the learning process and is thus critical to the quality of the forecasting. To this end, it is essential to ensure consistency between the target *metric* for forecasting and the employed *loss* function. In mobile network management, the relevant metric to assess the quality of the capacity allocation is the cost incurred by the operator, referred to as operator monetary cost (OMC). This metric captures the costs resulting from (i) forecasting a lower value than the actual offered load (which leads to the provisioning of insufficient resources); and (ii) predicting a higher value than the actual one (which leads to allocating more resources than those needed to meet the demand).

General-purpose loss functions like mean squared error (MSE) or mean absolute error (MAE) are clearly inappropriate to optimize the OMC. Indeed, these loss functions weigh all errors equally independent of whether the forecasting falls above or below the real value, and hence cannot learn the actual impact of different types of errors. Instead, a customized loss function is required to determine the actual penalty caused by a prediction error. In particular, by setting the loss function equal to the penalty inflicted by a given error in terms of OMC, the neural network is trained to minimize the metric of interest. In line with this, we design the loss function as follows:

- A constant penalty β is associated with each time slot where the allocated resources are lower than those needed in reality, leading to an SLA violation. This penalty value can be customized to the desired behavior. For example, higher values may be used for cases where reliability is needed, such as for ultra-reliable low-latency communications (URLLC) network slices. Instead, lower values can be applied for slices with more relaxed requirements.
- A monotonically increasing cost is attributed to resource overprovisioning, with a fixed rate of γ per overprovisioned byte. The more the resources (unnecessarily) provisioned, the higher the deployment cost for the operator. This reflects the deployment expenditure associated with excess allocated capacity, which we assume grows linearly with the amount of unused capacity. The linear scaling factor γ is configurable and represents the monetary cost of the excess resource allocation.

The configuration of the two cost models above can, in fact, be controlled by a single parameter α defined as the ratio between β and γ. Intuitively, α represents the amount of overprovisioned capacity that the operator is willing

to deploy to avoid committing an SLA violation. Operators can use α as a knob to steer the operational point of the system toward higher expenses in resource deployments but reduced chances of SLA violations, or vice versa.

The resulting loss function is flexible enough to accommodate different infrastructure locations (e.g., deploying resources at the network edge has a higher cost than at the core), resource types (e.g., radio resources are sensibly more expensive than CPU resources), and SLA strategies (e.g., slices providing critical services may entail higher violation fees).

Performance Evaluation

We evaluate the proposed framework with real-world data traffic recorded in the mobile network of a major European operator, providing coverage to a large metropolitan region. Our dataset includes information about the exchanged traffic of the most popular services, which we classify into seven categories (streaming, social network, web, cloud, gaming, messaging, and miscellaneous). It includes per-service traffic information provided as an aggregate over 5-minute intervals at 470 base stations. The data spans 11 weeks, of which we use 8 weeks for training, 2 for validation, and 1 for testing.

For the sole purpose of evaluating our algorithms with real traffic, we assume that each service category is assigned a dedicated slice, and adopt the methodology proposed in [14] to build a network topology model that associates network traffic to NFs and data centers. Our topology comprises different network levels ranging from the edge (the lowest level) to a fully centralized node (the highest level), such that higher network level nodes aggregate more traffic and serve a larger load. We refer to the highest level node as "core network data center" and the lowest level ones as "edge network data centers."

Unless otherwise stated, we fix T_p = 6 (which means that the forecasting modules are fed with data of the previous 30 minutes of traffic) and configure α = 1 (implying that one SLA violation has the same monetary cost as provisioning excess capacity sufficient to cover the traffic peak).

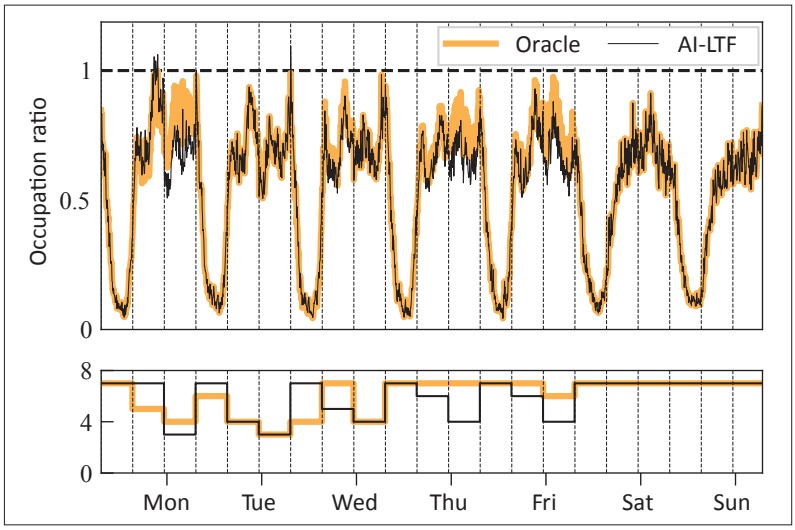

FIGURE 3. VNF placement of slices at one target data center. Occupation ratio (top) and number of admitted slices (bottom) for each 8-hour orchestration period. The algorithm implemented by the `AI-LTF` module is compared against an optimal but unfeasible oracle solution with perfect knowledge of the future traffic load.

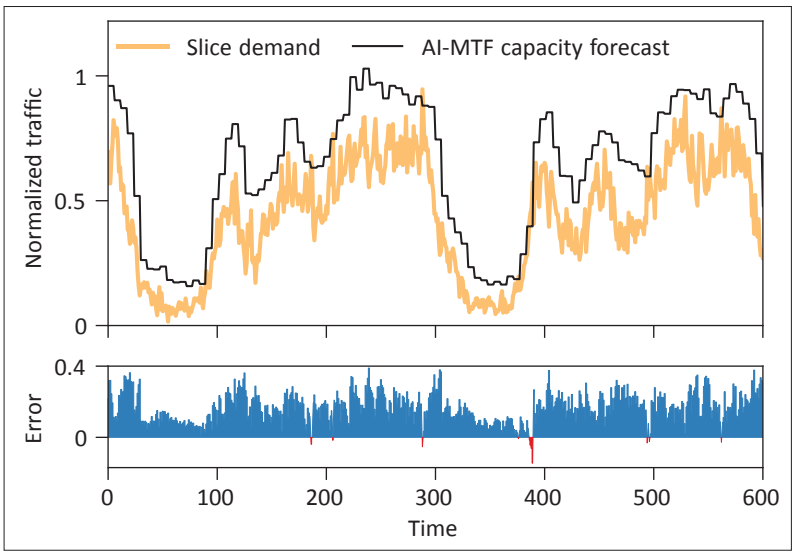

FIGURE 4. NFVI scaling for a slice serving streaming traffic at one target data center. Allocated capacity vs. service demand (top) and excess capacity (bottom) of `AI-MTF`. Values are normalized to the peak allocated capacity. Excess demand is shown in blue and unserviced demand in red.

`AI-LTF`: Long-Term Forecasting for VNF Placement

The long-term forecasting capabilities provided by the `AI-LTF` algorithm are useful to make decisions about the suitable placement of the VNFs serving one or more slices. To evaluate its performance, we consider a scenario where a data center with processing capacity C serves the seven slices and assume that the computational demand of a given slice is proportional to the amount of transmitted bytes.

In this case study, we set T_h = 8 hours to account for the fact that VNF placement decisions are typically taken with a coarse time granularity of hours due to the limitation of the underlying NFV technology. We focus on an edge network data center and employ `AI-LTF` to support the VNF placement decisions made by the NFVO module by anticipating the overall traffic load at the target data center. Then the NFVO can decide at every T_h how many slices are served by the data center of capacity C and which slices shall instead be placed elsewhere.

Figure 3 depicts the result obtained with `AI-LTF` against that obtained with an *oracle* algorithm that assists the NFVO with knowledge of the real future demand. (Such an oracle algorithm is unfeasible in practice but provides an optimal benchmark to assess `AI-LTF`'s performance.) We observe that `AI-LTF` follows the oracle quite closely. The overall usage of the deployed infrastructure remains high at all times. The algorithm only moves more slices than needed away from the data center on very limited occasions. In rare cases, it places more slices than it should in the data center, leading to an overload situation that results in computational outages for the served slices; however, even when this happens, the actual overload levels are negligible. These results confirm that `AI-LTF` is a promising solution to assist effective VNF placement decisions.

AI–MTF: Mid-Term Forecasting for NFVI Scaling

Once the VNFs serving various slices are placed at a given data center, it is possible to dynamically reallocate the resources assigned to each slice within the capacity C of the data center by scaling up or down the resources assigned to each slice. The time dynamics involved in such up- and downscaling are faster than those analyzed in the previous experiment for the VNF placement. Indeed, resource provisioning within the same data center (which involves booting up a VNF and setting up the data plane) can be performed at timescales of tens of minutes.

The AI–MTF algorithm can support this resource up- and downscaling process. We investigate its performance in a case study where the resources allotted to the slice serving streaming traffic at a core network data center are scaled every 30 minutes. Results, shown in Fig. 4, confirm that the proposed algorithm yields remarkable accuracy. The allocated capacity to the slice is scaled up and down to closely match the demand generated by the service. As highlighted in the bottom plot, the capacity allocated in excess is quite small, which implies that limited resources are wasted due to overprovisioning. Furthermore, the algorithm almost never incurs underprovisioning, and thus it always serves the offered demand and avoids violating the slice SLA.

AI–STF: Short-Term Forecasting for QoS Policies

The optimization of policies and resource allocations for individual flows or aggregates at different levels (PCF, RIC, RAN resource orchestration) can be performed at shorter timescales than those considered before. In particular, depending on the specific operation, these updates can be performed within intervals of a few minutes or less.

The AI–STF module is intended to back up this kind of high-pace network management task. We provide an example of application in Fig. 5 for the case of resource allocation, analyzing the network resources assigned to streaming flows in an edge network data center based on the prediction returned by AI–STF over time periods of T_h = 5 minutes (which is the finest time granularity available in our dataset). Specifically, the figure shows the distribution of the ratio of allocated resources to the demand, where a value below 1 denotes that the capacity forecast is not sufficient to satisfy the demand, while values above 1 mean that we have allocated more capacity than needed.

We observe that AI–STF is effective in provisioning sufficient resources to serve the aggregate demand for streaming flows while avoiding wasting too many resources in overprovisioning. We also observe that parameter α can be tuned to choose the desired trade-off between resource overprovisioning and SLA violations. Larger α values, corresponding to higher penalties for SLA violations, significantly reduce the probability of resource shortage, obviously at the cost of increasing the amount of resources wasted in overprovisioning.

Overall Performance

We next evaluate the overall performance of the three algorithms when jointly running in a complete 5G system. We consider the total load generated by the seven service categories

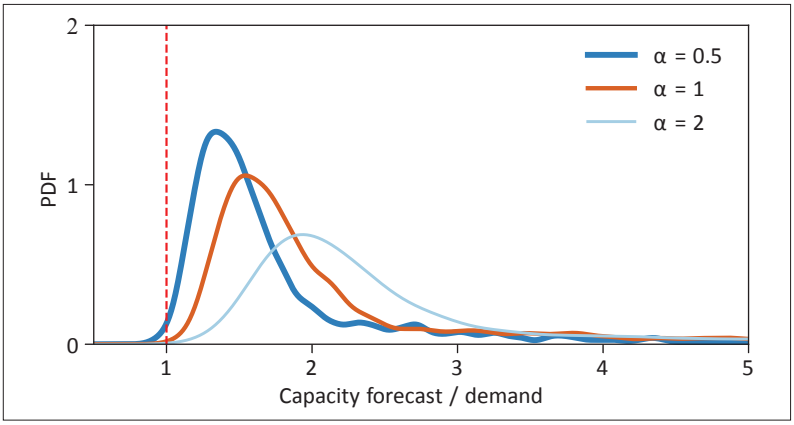

FIGURE 5. Distribution of the ratio of the allocated capacity with AI–STF over the aggregated demand of the streaming flows at a target edge network data center. Different curves correspond to diverse α ratios of the monetary penalty of SLA to the cost of overprovisioning. The integral of the curve for values of the abscissa below 1 corresponds to the probability of SLA violation.

	Unserviced demand (%)			Cost gains (%)		
	α = 0.5	α = 1	α = 2	α = 0.5	α = 1	α = 2
AI–LTF	0.53%	0.43%	0%	37.85%	56.28%	80.52%
AI–MTF	0.09%	0.08%	2.4e–3%	21.77%	64.4%	82.15%
AI–STF	8.5e–3%	4.8e–4%	3.4e–5%	23.33%	66.44%	81.43%
Overall system	0.63%	0.51%	2.4e–3%	31.04%	60.58%	81.09%

TABLE 1. Unserviced demand and cost gains for AI–LTF, AI–MTF, AI–STF, and the overall system, for different α values. The percentage of unserviced demand is given by the amount of traffic exceeding the capacity forecasted by AI–LTF, AI–MTF, and AI–STF. Cost gains are computed as the difference between the costs of the traditional and AI-based approaches over the cost of the traditional approach. The cost of the overall system is computed as the sum of the costs of the three algorithms.

at a core network data center where AI–LTF targets the aggregate load at the data center, while AI–MTF and AI–STF focus on the individual allocation for each service category. The results, given in Table 1, show the percentage of unserviced demand and the cost gains provided by our AI-based algorithms over a traditional forecasting technique, namely a seasonal autoregressive integrated moving average (ARIMA) [15].

The results on unserviced demand confirm the effectiveness of α in controlling the level of reliability at the expense of a larger resource deployment. Indeed, when selecting a sufficiently large α, we can achieve practically zero outages, which may be suitable to support, for example, URLLC services. Even for low values of α, the overall unserviced traffic remains reasonably low (below 1 percent). As expected, accuracy increases when the predicted time horizon is shorter (which explains why AI–STF outperforms AI–MTF for all αs and AI–MTF outperforms AI–LTF for α = 0.5 and α = 1) as well as when the traffic aggregate is larger (which explains why AI–LTF outperforms AI–LTF and AI–STF for α = 2).

The results on cost gains show the advantage of our approach over a seasonal ARIMA model [15]. In order to better align the seasonal ARIMA model with the requirements of the capacity fore-

casting problem, we augmented it with fixed overprovisioning on top of the predicted traffic; in line with benchmarks in the literature, we set an overprovisioning of 5 percent of the estimated peak traffic [9]. The results confirm that our algorithms attain much smaller OMCs than the traditional technique, with gains of up to 80 percent.

Conclusions

In this article, we present some of the challenges and opportunities that AI offers in the context of 5G networks. By defining a framework that joins contributions from various initiatives and populating it with AI-based algorithms serving different purposes, we show how standards can be leveraged to deploy AI-based 5G systems. Our performance evaluation results illustrate the benefits of proper integration of AI into 5G. Importantly, this work also provides a basis to apply AI to other functions within the 5G system beyond the ones addressed in the article.

Acknowledgment

This work was supported by the H2020 5G-TOURS European project (Grant Agreement No. 856950).}

References

[1] 3GPP TS 23.288 v16.1.0, "Architecture Enhancements for 5G System (5GS) to Support Network Data Analytics Services (Release 16)," June 2019.
[2] 3GPP TS 28.533 v16.0.0, "Management and Orchestration of Networks and Network Slicing; Management and Orchestration Architecture (Release 16)," June 2019.
[3] O-RAN Alliance White Paper, "O-RAN: Towards an Open and Smart RAN," Oct. 2018.
[4] ETSI White Paper No. 32, "Network Transformation (Orchestration, Network and Service Management Framework)," Oct. 2019.
[5] 3GPP TR 28.890 v16.0.0, "Study on Integration of Open Network Automation Platform (ONAP) and 3GPP Management for 5G Networks (Release 16)," Mar. 2019.
[6] J. Wang *et al.*, "Spatiotemporal Modeling and Prediction in Cellular Networks: A Big Data Enabled Deep Learning Approach," *Proc. IEEE INFOCOM*, Atlanta, GA, May 2017.
[7] C. Zhang and P. Patras, "Long-Term Mobile Traffic Forecasting Using Deep Spatio-Temporal Neural Networks," *Proc. ACM Mobihoc*, Los Angeles, CA, June 2018.
[8] C. Marquez *et al.*, "Not All Apps Are Created Equal: Analysis of Spatiotemporal Heterogeneity in Nationwide Mobile Service Usage," *Proc. ACM CoNEXT*, Seoul/Incheon, South Korea, Nov. 2017.
[9] D. Bega *et al.*, "Deepcog: Optimizing Resource Provisioning in Network Slicing with Ai-Based Capacity Forecasting," *IEEE JSAC*, vol. 38, no. 2, Feb. 2020, pp. 361–76.
[10] K. Hara, H. Kataoka, and Y. Satoh, "Can Spatiotemporal 3D CNNs Retrace the History of 2D CNNs and ImageNet?" *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, Salt Lake City, UT, Nov. 2018.
[11] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving Deep Neural Networks for LVCSR Using Rectified Linear Units and Dropout," *Proc. IEEE ICASSP*, Vancouver, Canada, May 2013.
[12] M. W. Gardner and S. R. Dorling, "Artificial Neural Networks (The Multilayer Perceptron) — A Review of Applications in the Atmospheric Sciences," *Atmospheric Environment*, vol. 32, no. 14–15, Aug. 1998, pp. 2627–36.
[13] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," Dec. 2014;. arXiv:1412.6980.
[14] C. Marquez *et al.*, "How Should I Slice My Network?: A Multi-Service Empirical Evaluation of Resource Sharing Efficiency," *Proc. ACM MobiCom*, New Delhi, India, Nov. 2018.
[15] R. J. Hyndman and G. Athanasopoulos, *Forecasting: Principles and Practice*, 2nd ed., Otexts, 2018.

Biographies

Dario Bega was at IMDEA Networks at the time this article was written and is currently a core network research specialist at Nokia Bell Labs, Munich. He received his Ph.D. from University Carlos III of Madrid (UC3M), Spain, in 2020 and his M.Sc. degree in telecommunication engineering from the University of Pisa, Italy, in 2013. The views expressed in this article do not reflect the position of Nokia Bell Labs.

Marco Gramaglia received his M.Sc. and Ph.D. degrees in telematics engineering from UC3M in 2009 and 2012, respectively. He held postdoctoral research positions at ISMB, Italy, CNR-IEIIT, Italy, and IMDEA Networks. Currently, he is at UC3M. He has been involved in several European projects, playing key roles such as Deputy Technical Manager or WP leader, and has authored more than 50 papers published in international conferences and journals.

Ramon Perez is a Ph.D. candidate at UC3M and works at Telcaria Ideas S.L., a startup focusing on advanced network virtualization solutions in Spain. He received his B.Sc. in telecommunication technology engineering from the University of Seville in 2015 and his M.Sc. in telecommunication engineering from the Polytechnic University of Madrid in 2017. His research interests include network slicing, network virtualization, monitoring, and network automation in 5G networks.

Marco Fiore is a research associate professor at IMDEA Networks Institute, Spain. He received his M.Sc. degrees from the University of Illinois at Chicago and Politecnico of Torino, Italy, a Ph.D. degree from Politecnico di Torino, Italy, and a Habilitation à Diriger des Recherches from Université de Lyon, France. He was a researcher at Consiglio Nazionale delle Ricerche (CNR), Italy, an associate professor at Institut National des Sciences Appliquées (INSA) de Lyon, France, and a visiting researcher at Rice University, Texas, Universitat Politècnica de Catalunya, Spain, and University College London, United Kingdom.

Albert Banchs has a double affiliation as professor at the University Carlos III of Madrid and deputy director of IMDEA Networks Institute. He is an author of more than 100 publications, has been Principal Investigator of 9 European Projects, and has served on many TPCs and Editorial Boards. Currently, he is Deputy Technical Manager of the 5G-TOURS project and serves on the Editorial Board of *IEEE/ACM Transactions on Networking*. He received his M.Sc. and Ph.D. degrees from the Polytechnic University of Catalonia (UPC) in 1997 and 2002.

Xavier Costa-Pérez is head of Beyond 5G Networks R&D at NEC Laboratories Europe, scientific director at the i2Cat R&D Center, and a research professor at ICREA. His team contributes to products roadmap evolution as well as to European Commission R&D collaborative projects and received several awards for successful technology transfers. He has served on the Program Committee of several conferences (including IEEE Greencom, WCNC, and INFOCOM), published at top research venues, and holds several patents. He received his M.Sc. and Ph.D. degrees in telecommunications from the Polytechnic University of Catalonia in Barcelona.