AI-based Autonomous Control, Management, and Orchestration in 5G: from Standards to Algorithms

Dario Bega*†, Albert Banchs†*, Marco Gramaglia†, Marco Fiore*, Ramon Perez‡†, Xavier Costa-Perez§
*IMDEA Networks Institute, Spain
†University Carlos III of Madrid, Spain
‡Telcaria Ideas, Spain
§NEC Laboratories, Germany

***Abstract*—While the application of Artificial Intelligence (AI) to 5G networks has raised a 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 paper, we contribute to bridging the gap between standards and a working solution, by defining a framework that brings together the relevant standards specifications and complements them with the required missing building blocks. We populate this framework with concrete AI-based algorithms that serve different purposes towards 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.**

I. 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 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: the coexistence of heterogeneous mobile services, diversified network requirements, and tenant-defined management policies creates a need for specialized and time-varying infrastructure deployments, which in turn call for automated solutions in the control, management, and orchestration of the network.

Artificial Intelligence (AI) is a natural choice to support the emerging need in autonomous network operation and management. 3GPP and other Standard Developing 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 towards 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 into its standardized architecture the modules (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]. 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 into 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 paper is to complement and support ongoing standardization activities by developing and populating a unified framework that leverages data analytics and AI for network control, management and orchestration. More precisely, we set forth the following main contributions:

- We propose a comprehensive framework for the incorporation of data analytics and AI in traditional network architectures, which brings together the corresponding efforts at relevant standardization bodies like 3GPP and ETSI and complements them with additional modules that are needed to provide the desired functionality.
- We populate the proposed framework with practical algorithms that leverage AI and machine learning (ML) solutions to assist different types of control and orchestration decisions, namely (i) decisions on the most appropriate placement for the different VNFs, (ii) decisions on the scaling of VNF resources at run-time, and (iii) decisions on the adjustment of flow-level QoS parameters.
- We evaluate the performance of the proposed algorithms, providing results on their accuracy and showing their ability to effectively attain precise control, management and orchestration decisions at different timescales.

II. AI-driven Data Analytics framework

Figure 1 depicts the network data analytics framework we propose.[1] The framework design encompasses the Management and Orchestration plane as well as the Control plane functionalities, as AI can indeed improve the performance at

[1]While the figure shows functional interactions across modules, the different functions may actually be connected through message buses, as mandated by recent versions of the 3GPP standards [6].

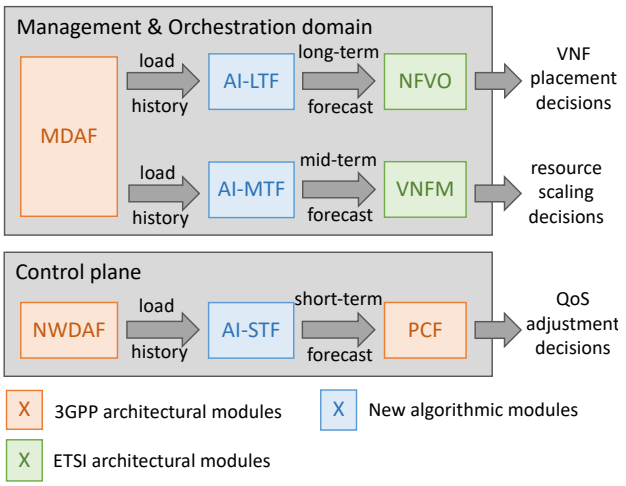

Fig. 1: Proposed framework with standard functions (MDAF, NFVO, VNFM, NWDAF, PCF) and new modules implementing AI-based algorithms (AI−LTF, AI−MTF, AI−STF).

all levels. Within each plane, we take as reference architecture the one proposed by 3GPP and integrate it with an ETSI NFV MANO architecture. The resulting design is also well aligned with other emerging architectures such as the aforementioned ETSI ENI and ZSM, and O-RAN. The main novelties of the proposed framework are as follows: (*i*) we bring together modules from 3GPP and ETSI standards, (*ii*) we integrate them with new modules implementing the AI algorithms, and (*iii*) we apply the framework to different purposes going beyond the standards.

A. 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 slices instances, sub-instances and network functions hosted within a network infrastructure. This includes the centralized collection of network data for subsequent publishing to other network management and orchestration modules. In the proposed framework, we specifically employ the MDAF to collect mobile data traffic loads generated in the radio access domain by the individual slices or flows. As a result, the MDAF allows building historical databases of the network demands for each base station or sector. These data are then exposed to the AI-based prediction algorithms for (*i*) long-term forecasting (AI−LTF), and (*ii*) mid-term forecasting (AI−MTF).

The AI−LTF algorithm aims at assisting the VNF placement decisions taken 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 such a prediction against the current available capacity in each infrastructure location and anticipates potential overload conditions. The NFVO can react, *e.g.*, by moving VNFs out of the congested infrastructure. The AI−LTF algorithm operates on long timescales, typically in the order of hours: indeed, VNFs 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 taken 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 on the level of the unsatisfied demand and the amount of unused resources. Based on all this information, the AI−MTF algorithm assists the orchestration framework on the decision (*i*) 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 (*ii*) to *downscale* resources to save cost when VNFs are leaving a significant fraction of the resources unused. Such decisions must be taken over faster timescales than those affecting the VNF placement, and generally occur over intervals in the order of tens of minutes, which is the typical frequency for the execution of new VNF instances involving upscaling and downscaling.

B. Control plane

On the control plane, the NWDAF module is responsible for collecting data on the network load, 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 of flows (or flow aggregates). The forecast is leveraged by the Policy Control Function (PCF) module, which provides a unified policy framework to govern the network behavior (*e.g.*, dealing with radio or transport network resources). In particular, PCF can adjust the QoS parameters associated to the different flows in advance, based on predicted demands, so as to guarantee better QoS support. These operations are performed at rather fast timescales, in the order of minutes or less, as QoS parameters can be frequently updated without incurring substantial overhead.

While the NWDAF modules have been designed for the network core, a similar approach could be applied to the radio access network (RAN). Indeed, although 3GPP has not yet proposed modules equivalent to NWDAF in the RAN, other initiatives such as the O-RAN alliance have defined elements such as the RAN Intelligent Controller (RIC), which can collect and distribute data at the RAN level. Similarly to our approach above, the RIC can act on QoS parameters at the base station level. Decisions become then local, and can be taken much faster, hence tracking rapid traffic dynamics and increasing the radio access management efficiency.

III. AI-BASED ALGORITHMS DESIGN

The above framework introduces three new AI-based algorithmic modules: AI−LTF, AI−MTF and AI−STF. The three algorithms follow the same design guidelines, as all of them aim at providing network capacity forecasts. The main difference between them is that they work at different granularities in terms of traffic volumes (at global, slice, or flow levels) and timescales (hours, tens of minutes, or

minutes). In the following, we present the unified design of these three algorithms.[2]

A. Capacity forecasting

Given the complexity of predicting network traffic, our algorithm design takes advantage of recent advances in supervised learning via Deep Neural Network (DNN) architectures, which are well suited to deal with the high input data complexity associated with spatiotemporal fluctuations in mobile data traffic [7]. For this reason, DNN-based solutions have recently gained momentum in network management research. Yet, in contrast to the majority of the literature in the field, our DNN 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 minimizing legacy cost functions such as Mean Square Error (MSE) or Mean Absolute Error (MAE); in other words, they try to match the temporal behavior of traffic, giving the same weight to positive and negative errors [8]. While this produces forecasts that reduce as much as possible the error between the future and the anticipated demand, this approach 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 SLA violations that have a monetary penalty much higher than the cost resulting from overdimensioning the resources, as long as such 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 build on recent proposals that properly model the monetary costs incurred by the mobile network operator [9], [10], adjusting their design to the particular requirements of the proposed framework.

B. Algorithm design overview

The algorithm design 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 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 and NWDAF for each slot. Such 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 to datacenters such that a datacenter serves the aggregated load of all the associated bases stations. We consider different levels of datacenters, ranging from the first level, where we have a different micro-datacenter co-located with each base station, to the last level, where we have a single large datacenter serving the entire network. Our framework aims at allocating the required capacity at all datacenters or associated network functions. Typically, `AI-STF` works at levels close to the edge, while `AI-LTF` and `AI-MTF` may operate at all levels.

Our goal is to compute a *constant* capacity to be allocated in the network datacenters 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.

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 (i) traffic loads with diverse demand patterns, (ii) any datacenter level, and (iii) any time horizon T_h. This allows to leverage the same DNN design to implement all three algorithmic modules. 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. 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 datacenter class 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 loss function to measure the quality of the forecasting and steer the system over the training phase. Our loss function targets the overall (monetary) metric rather than a generic one, considering the compound cost of overprovisioning and underprovisioning network resources when allocating a constant capacity over the time horizon to serve the actual time-varying demand.

In the remainder of this section, we detail the implementation of the above three components.

C. Encoder and decoder structure

The neural network architecture used by the proposed modules is summarized in Figure 2, and is composed of an encoder-decoder sequence. While the three algorithms considered in this paper (`AI-LTF`, `AI-MTF`, and `AI-STF`)

[2]Implementation available at `https://github.com/wnluc3m`.

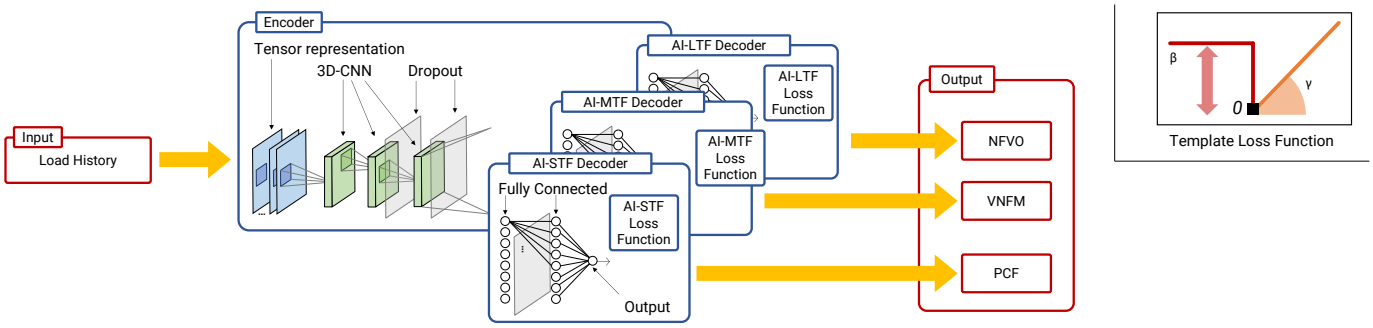

Fig. 2: Neural network encoder-decoder structure.

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.

The internal structures of the encoder and decoder are inspired by recent breakthroughs in deep learning for image and video processing [11]. 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 [11]. Convolutional Neural Networks (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 or image quality assessment. 3D-CNNs extend 2D-CNNs to address the case were 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 $3\times3\times3$ kernel for the first 3D-CNN layer and a $6\times6\times6$ 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 [12].

The decoder uses Multi-Layer Perceptrons (MLPs) [13], 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 datacenters).

For the training procedure, we employ the popular Adam optimizer, which is a Stochastic Gradient Descent (SGD) method providing fast convergence [14]. 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.

D. Loss function design

The loss function drives the learning process and is thus critical to the quality of the forecasting. In mobile network management, the relevant metric to assess the quality of the capacity allocation is the Operator Monetary Cost (OMC). Hence, the loss function has to reflect the difference between the capacity forecast and the actual demand in terms of OMC.

General-purpose loss functions like MSE or MAE are clearly inappropriate to this end, and a customized loss function is required to determine the actual penalty caused by a prediction error. Such penalty corresponds to 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). These costs are as follows:

- A constant penalty β is associated to each time slot where the allocated resources are lower than those needed in reality, leading to an SLA violation. Such penalty value can be customized to the desired behavior, *e.g.*, higher values may be used for cases where reliability is needed, *e.g.*, in 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 that 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 before committing an SLA violation. Operators can use α as a knob to steer the operational point of the system towards higher expenses in resource deployments but reduced chances of SLA violations, or vice-versa. We provide examples of α parametrization in Section IV.

The resulting loss function is flexible enough to accommodate different infrastructure deployment 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). Furthermore, it can be parameterized to account for the overall cost over different time intervals as required by the different algorithms (AI−LTF, AI−MTF, and AI−STF).

IV. 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 seven popular services (including, among others, Youtube, Facebook, and Whatsapp), with per-service traffic information provided as an aggregate over 5-minute intervals at 470 4G base stations. The data spans 11 weeks, where we use 8 weeks for training, 2 for validation and the remaining one for testing.

We assume that each service is assigned a dedicated slice, and adopt the methodology proposed in [15] to build a network topology model that associates base stations to edge and core network datacenters. 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), configure $\alpha = 1$ (implying that one SLA violation has the same monetary cost as provisioning an excess capacity sufficient to cover the traffic peak) and focus on a core network datacenter.

A. AI−LTF: Long-term forecasting for VNF placement

The long-term forecasting capabilities provided by the AI−LTF module are useful to make decisions about the suitable placement of the VNFs serving one or more slices. To evaluate the performance of this module, we consider a scenario where a datacenter with processing capacity C must serve the seven slices and assume that the computational demand of a given slice is proportional to the amount of bytes demanded by the corresponding service.

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 datacenter and employ AI−LTF to support the VNF placement decisions taken by the NFVO module by anticipating the overall traffic load at the target datacenter. Then, the NFVO can decide at every T_h how many slices are served by the datacenter 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 the knowledge of the real future demand (even though such an oracle algorithm is unfeasible in practice, it does provide an optimal benchmark to assess AI−LTF's performance). We observe that AI−LTF follows quite closely the oracle. The overall usage of the deployed infrastructure remains high at all times. The algorithm only moves more slices than needed away from the datacenter in very limited occasions. In rare cases, it places more slices than it should in the datacenter, leading to overload situation that results into 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.

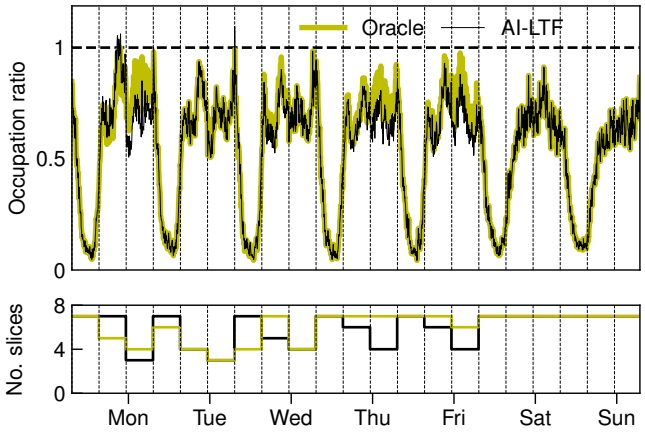

Fig. 3: VNF placement of slices at one target datacenter. Occupation ratio (top) and number of admitted slices (bottom) for each 8-hour orchestration period. The algorithm implemented by the AI−LTF module is compared against an optimal but unfeasible oracle solution with perfect knowledge of the future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

B. AI−MTF: mid-term forecasting for NFVI scaling

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

The AI−MTF module can support such resource up and down scaling process. We investigate its performance in a case study where the resources allotted to the slice serving Youtube traffic at a datacenter are scaled every 30 minutes. Results, shown in Figure 4, confirm that the proposed algorithm yields remarkable accuracy. The allocated capacity to the slice is scaled up and down to match closely 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.

C. AI−STF: short term forecasting for QoS enforcement

The allocation of network resources and the setting of QoS parameters for individual flows or aggregates can be adjusted at shorter timescales than those considered before. Indeed,

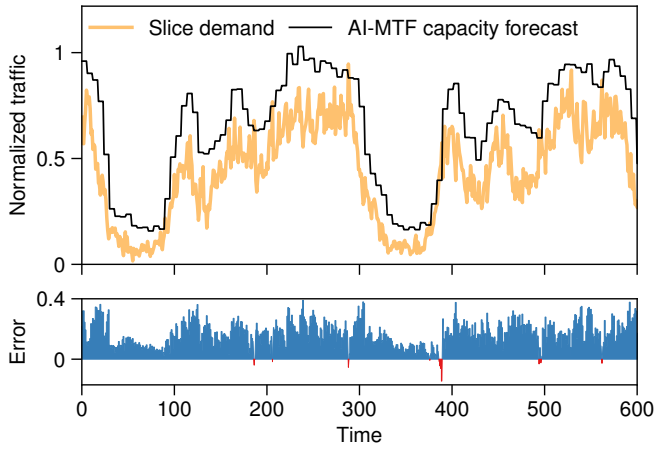

Fig. 4: NFVI scaling for a slice serving Youtube traffic at one target datacenter. Allocated capacity versus service demand (top) and excess capacity (bottom) of AI−MTF. Values are normalized to the peak allocated capacity. Excess demand is shown in blue and unserviced demand in red.

the enforcement of QoS policies or the configuration of QoS mechanisms allows adapting the amount of resources assigned to flows within intervals of a few minutes or less.

The AI−STF module is intended to back up this kind of high-pace network management tasks. We provide an example of application in Figure 5 for the case of resource allocation, analyzing the network resources assigned to Youtube flows in the network edge based on the prediction returned by AI−STF over time periods of $T_h = 5$ minutes. Specifically, the figure shows the distribution of the ratio of assigned 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 allocated more capacity than needed.

We observe that AI−STF is effective in provisioning sufficient resources to serve the aggregate demand for Youtube flows while avoiding wasting too many resources in overprovisioning. We also observe that the 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, reduce significantly the probability of underprovisioning, obviously at the cost of increasing the amount of resources wasted in overprovisioning (*i.e.*, shifting the distribution to the right).

D. 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 services at a cloud network datacenter and compute the percentage of unserviced demand as given by the amount of traffic exceeding the capacity forecasted by AI−LTF, AI−MTF, and AI−STF, respectively. Following the framework in Section II, AI−LTF targets the aggregate load at the datacenter, while AI−MTF and AI−STF focus on the individual allocation for each service.

The results, given in Table I for different values of α, confirm the effectiveness of α in reducing the amount of

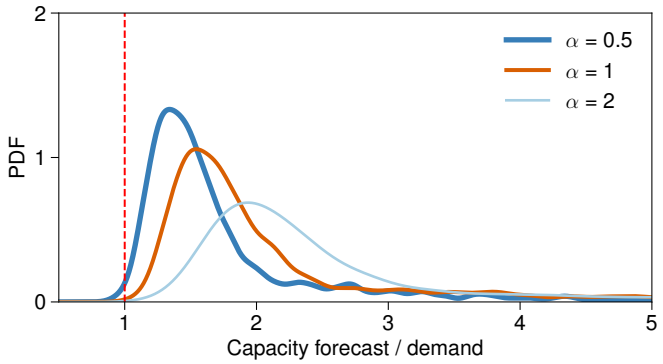

Fig. 5: Distribution of the ratio of the allocated capacity with AI−STF over the aggregate demand of the Youtube flows at a target edge network datacenter. Different curves correspond to diverse α ratios of the monetary penalty of SLA to the cost of overprovisioning. The integral of the curve for values of the abscissa below 1 corresponds to the probability of SLA violation.

TABLE I: Percentage of unserviced demand caused by the capacity predictions of the AI−LTF, AI−MTF, and AI−STF modules, and in the overall system combining the three algorithms, for different α values.

Unserviced demand (%)	$\alpha = 0.5$	$\alpha = 1$	$\alpha = 2$
AI-LTF	0.53 %	0.43 %	0 %
AI-MTF	0.09 %	0.08 %	2.4e-3 %
AI-STF	8.5e-3 %	4.8e-4 %	3.4e-5 %
Overall system	**0.63 %**	**0.51 %**	**2.4e-3 %**

unserviced demand 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, *e.g.*, URLLC services. Even for low values of α, the overall unserviced traffic remains reasonably low (below 1%). We further observe that, as expected, accuracy increases when the predicted time horizon is shorter (which explains why AI−STF outperforms AI−MTF for all α's and AI−MTF outperforms AI−LTF for $\alpha = 0.5$ and $\alpha = 1$) as well as when the traffic aggregate is larger (which explains why AI−LTF outperforms AI−LTF and AI−STF for $\alpha = 2$).

In summary, these results further corroborate the effectiveness of an integrated AI framework for control, management, and orchestration of a 5G sliced network system.

V. Conclusions

In this paper, we presented some of the challenges and opportunities that AI offers in the context of 5G networks. By defining a framework that joins contributions from different SDOs, populating it with different AI-based algorithms, and applying it for different purposes, we showed how standards can be leveraged to deploy AI-based 5G systems. Our performance evaluation results illustrate the benefits of a 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 paper.

REFERENCES

[1] 3GPP TS 23.288 v16.1.0, "Architecture Enhancements for 5G System (5GS) to Support Network Data Analytics Services (Release 16)," Jun. 2019.

[2] 3GPP TS 28.533 v16.0.0, "Management and Orchestration of Networks and Network Slicing; Management and Orchestration Architecture (Release 16)," Jun. 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] 3GPP TS 23.501 v16.2.0, "System Architecture for the 5G System (Release 16)," Sep. 2019.

[7] C. Marquez, M. Gramaglia, M. Fiore, A. Banchs, C. Ziemlicki, and Z. Smoreda, "Not All Apps Are Created Equal: Analysis of Spatiotemporal Heterogeneity in Nationwide Mobile Service Usage," in *Proc. of ACM CoNEXT*, Nov. 2017.

[8] M. Wang, Y. Cui, X. Wang, S. Xiao, and J. Jiang, "Machine Learning for Networking: Workflow, Advances and Opportunities," *IEEE Network*, vol. 32, no. 2, pp. 92–99, Mar. 2018.

[9] 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.

[10] ——, "DeepCog: Optimizing Resource Provisioning in Network Slicing with AI-based Capacity Forecasting," *IEEE Journal of Selected Areas in Communications*, to appear.

[11] K. Hara, H. Kataoka, and Y. Satoh, "Can spatiotemporal 3D CNNs retrace the history of 2D CNNs and ImageNet?" in *Proc. of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, Nov. 2018, pp. 191–206.

[12] G. E. Dahl, T. N. Sainath, and G. E. Hinton, "Improving deep neural networks for LVCSR using rectified linear units and dropout," in *Proc. of IEEE ICASSP*, Vancouver, Canada, May 2013, pp. 8609–8613.

[13] 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, pp. 2627–2636, Aug. 1998.

[14] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv:1412.6980*, Dec. 2014.

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