Papagianni, C., Mangues-Bafalluy, J., Bermudez, P., Barmpounakis, S., ... Pepe, T. (2020). 5Growth: AI-driven 5G for Automation in Vertical Industries. In *Proceedings of 2020 European Conference on Networks and Communications (EuCNC)*.

DOI: https://doi.org/10.1109/EuCNC48522.2020.9200919

5Growth: AI-driven 5G for Automation in Vertical Industries

Chrysa Papagianni[*], Josep Mangues-Bafalluy[†], Pedro Bermudez[‡], Sokratis Barmpounakis[§], Danny De Vleeschauwer[*], Juan Brenes[¶], Engin Zeydan[†], Claudio Casetti[‖], Carlos Guimarães[**], Pablo Murillo[‡], Andres Garcia-Saavedra[††], Daniel Corujo[‡‡], Teresa Pepe[x]
[*]Nokia Bell Labs, [†]Centre Tecnològic de Telecomunicacions de Catalunya, [‡]Telcaria,
[§]National and Kapodistrian University of Athens, [¶]Nextworks, [‖]Politecnico di Torino, [**]Universidad Carlos III de Madrid,
[††]NEC Laboratories Europe, [‡‡]Instituto de Telecomunicações e Universidade de Aveiro, [x]Ericsson Research

***Abstract*—Spurred by a growing demand for higher-quality mobile services in vertical industries, 5G is integrating a rich set of technologies, traditionally alien to the telco ecosystem, such as machine learning or cloud computing. Despite the initial steps taken in prior research projects in Europe and beyond, additional innovations are needed to support vertical use cases. This is the objective of the 5Growth project: automate vertical support through (i) a portal connecting verticals to 5G platforms (*a.k.a.* vertical slicer), (ii) closed-loop machine-learning based Service Level Agreement (SLA) control, and (iii) end-to-end optimization. In this paper, we introduce a set of key 5Growth innovations supporting radio slicing, enhanced monitoring and analytics and integration of machine learning.**

I. Introduction

In addition to performance and radio-related enhancements, 5G has also paved the way for the innovative integration of mechanisms coming from other technological domains such as cloud computing, machine learning and service-based architectures. Such integration enables for the first time to cater for the highly heterogeneous requirements on service provisioning towards verticals. The development of such integrated mechanisms becomes paramount to operators, manufacturers and service providers, second only to the need for validation and experimentation alongside the verticals themselves. Projects such as H2020 5G-TRANSFORMER have provided decisive first steps towards that direction, exploiting technologies such as Network Function Virtualization, Software Defined Networking and advances in service orchestration, amongst others, to partition the network in different slices addressing different communication needs from disparate verticals. Despite this important initial step, there is also the need to assess the capability of such mechanisms to encompass not only key performance targets directly at verticals' premises, but also to support automation and optimisation of end-to-end connectivity solutions. This is the objective of the 5Growth project which, besides deploying such capabilities in advanced field trials alongside verticals, adds extensions and innovative capabilities to 5G platforms. In this paper, we present design considerations and preliminary results for a key set of such innovations and associated extensions, including network monitoring and analytics, slicing at the radio access network, machine-learning (ML) based resource allocation and user profiling supporting smart orchestration.

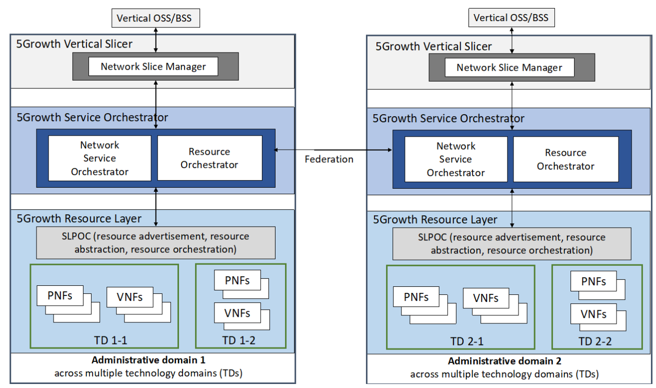


Fig. 1: 5Growth Baseline Architecture

II. Baseline Platform

The 5Growth architecture, depicted in Fig. 1, builds onto the one of 5G-TRANSFORMER [1], with enhancements along the following dimensions: usability, flexibility, automation, performance and security. The goal is to enable uniform and automated deployment and operation of slices customized with requirements from the vertical industries in the project, spanning from Industry 4.0 to Transportation and Energy. The architecture is composed by three core building blocks: 5Growth Vertical Slicer (5Gr-VS), 5Growth Service Orchestrator (5Gr-SO) and 5Growth Resource Layer (5Gr-RL), in addition to components related with the monitoring and decision automation to support the former blocks.

A. 5Growth Vertical Slicer (5Gr-VS)

The 5Gr-VS, extending the 5G-TRANSFORMER Vertical Slicer, acts as a one-stop shop entry point for verticals requesting the provisioning and management of vertical services through a simplified and vertical-oriented northbound interface with the vertical OSS/BSS.

Through this northbound interface, each vertical can request a vertical service by initially selecting a "template" from the catalogue of Vertical Service Blueprints (VSB), to be used as basis for the service definition. Then verticals can complete their specification providing a number of service-oriented parameters that customize the desired service instance. The goal of this approach is to enable the verticals to focus only on the requirements, the high-level components and the logic of their service applications and their relation. The actual deployment of all network-related components (e.g., for mobile connectivity) and underlying resource handling is left to the lower layers of the 5Growth stack. The final specification of the vertical service, provided by the vertical, is formally expressed through a Vertical Service Descriptor (VSD), which is composed by the VSB filled with the user-defined parameters.

Afterwards, the 5Gr-VS handles the requests for vertical services by internally managing the mapping and translation between the requested vertical services (Vertical Service Instances - VSI) and a number of network slices (Network Slice Instances – NSI), which are then created on-demand by provisioning the underlying NFV network services (NFV-NS), requested to the 5Growth Service Orchestrator.

B. 5Growth Service Orchestrator (5Gr-SO)

The 5Gr-SO, which is inherited from the 5G-TRANSFORMER Service Orchestrator, provides both network service and resource orchestration capabilities to: (i) enabling end-to-end orchestration of NFV-Network Services (NFV-NS) by mapping them across a single or multiple administrative domains based on service requirements and availability of the services/resources offered by each of the domains; and (ii) managing their life-cycles (including on-boarding, instantiation, update, scaling, termination, etc.). In addition, the 5Gr-SO coordinates and offers the 5Gr-VS an integrated view of the services, which may be running in the local or in peer administrative domains. However, this is transparent to the 5Gr-Vs.

The 5Gr-SO receives the service requirements from the 5Gr-VS via its northbound interface in the form of Network Service Descriptors (NSD), which describe a NFV-NS and are expressed in terms of chaining of VNF components as well as their individual requirements. Additional components (e.g., monitoring jobs, scaling rules) may come in the request as well. Internally, the 5Gr-SO decides the optimal service (de)composition for the whole NFV-NS based on service availability as well as the resource capabilities exposed by the local 5Gr-RL and by other administrative domains as well as the optimal placement of VNFs and vertical applications (VAs) along with the optimal deployment of virtual links connecting VNFs through mapping operations over the topology exposed by the 5Gr-RL. The 5Gr-SO may request network services to federated 5Gr-SOs to address the execution of portions of the NFV-NS in other administrative domains. Additionally, it performs the lifecycle management of the whole NFV-NS, including nested NSs and each VNF composing the NFV-NS. Finally, it performs monitoring tasks and SLA management functions to enable the triggering of self-adaptation actions (e.g., healing and scaling operations), thereby preventing service performance degradation or SLA violations.

C. 5Growth Resource Layer (5Gr-RL)

The 5Gr-RL, which is inherited from 5G-TRANSFORMER Mobile Transport and Computing Platform, is responsible not only for managing the local infrastructure and the required transport resources, but also for resource orchestration and VNFs instantiation over the infrastructure under its control, as well as for managing the underlying physical transport network, computing and storage infrastructure. In other words, it hosts all the compute, storage and networking physical and virtual resources where the components of network slices and end-to-end services are executed.

The 5Gr-RL creates an abstract view of the available resources (e.g., transport (WIM), data-centers (VIM), RAN) and exposes it to the 5Gr-SO, based on which placement decisions are taken by the 5Gr-SO. The 5Gr-RL receives service requests from 5Gr-SO via its northbound interface. Based on the specific request, it maps the selected abstracted resources in the corresponding physical resources and allocates them. According to the abstraction algorithm, the 5Gr-RL can do the selection and optimization of resources using dedicated and specialized resource Placement algorithm which acts on the non-abstracted infrastructure resources. This is further elaborated in the smart orchestration section.

III. Architecture Innovations

The main architectural innovations of 5Growth revolve around two main issues, namely RAN support (including the interface exposed to the vertical and implications for all blocks) and the addition of intelligence for decision making, including the required monitoring framework.

A. Radio Access Network support in Vertical Slicer

A typical vertical service (VS) is traversal to the operator's network, since it needs to connect the end users with the service logic that can be arbitrarily placed at any place of the network. For this reason end-to-end network slices supporting VSs typically span from the Radio Access Network (RAN) to the core. These two segments have different characteristics and the way to model and allocate resources is radically different. On one side, the core segment of the network slice is usually composed using a number of Network Services (NSs) that define the Network Functions and the internal connection among them. On the other side, network slices need a specific set of network resources at the access segment that are tightly coupled to the mobile traffic profile of the VS. The decomposition of network slices into network services used in the core segment has already been addressed in the 5G-TRANSFORMER project, but the modelling and the configuration of the access segment was out of the scope of the project. In 5Growth we propose extensions to enable network slices encompassing core and access networks to support vertical services with end-to-end QoS and SLA guarantees.

The cornerstone for all the required extensions is the inclusion of mobile traffic profile and access network information as part of the Network Slice Template information model. This approach is fully aligned with the 3GPPP approach established in [2]. The idea is to include the parameters that characterize the specific service type (i.e. eMBB, URLLC, MTC) and some common parameters such as the coverage area, the required latency, etc. In this model, network slices are also composed

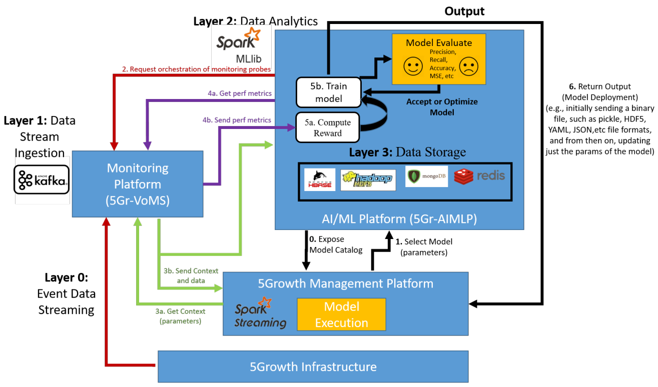


Fig. 2: 5Growth AI/ML workflow

by a set of Network Slice Subnets which are in turn composed by Network Services. This core and access combined network slicing approach renders the demarcation border between core and access segment functionalities more flexible, and even allows to move the traditional 5G-Core network functions towards the access for in-premises deployments.

In order to profit from this enhanced network slice templates, the 5G-TRANSFORMER Vertical Service Blueprint (VSB) and Vertical Service Descriptors (VSDs) information models also have be extended. On the one hand, the new VSB shall include a parameter to establish the desired service type and some high-level parameters to determine the default range that shall be guaranteed by the network slice. On the other hand, the VSD shall also be extended to allow to override the default values established for each specific service type parameter. 5Gr-VS will translate these new VSBs and VSDs into a network slice containing the access segment resources and the network services required to support the vertical service. From an architectural perspective, 5Gr-VS can rely on the interface between between 5GT-VS and the 5GT-SO for life-cycle management of the network services, but extensions are required in order to configure the radio access resources using the information available from the network slice. With this regard, the interface between the 5Gr-RL and the 5Gr-SO will be an evolution of 5G-TRANSFORMER's 5GT-SO-MTP interfaces since the 5Gr-RL should now expose an abstraction of the RAN infrastructure and provide RAN control primitives.

B. *AI/ML Platform (5Gr-AIMLP)*

3GPP has acknowledged the significance of data analytics for future cellular systems. In particular, a new function, called NetWork Data Analytics Function (NWDAF) has been introduced in [3]. NWDAF is responsible for providing network analysis information upon request from network functions, e.g., assisting the Policy Control Function (PCF) in selecting traffic steering policies. 5Growth is generalizing this idea by extending 5G-TRANSFORMER to integrate NWDAF concepts and provide an AI/ML platform for smarter control [4].

A workflow of the platform is depicted in Fig. 2, with two main functional blocks assisting the 5Growth platform (Fig. 1): the 5Gr-AIMLP—assisting in functions common to many AI/ML schemes, such as neural network fitting—and the 5Growth Vertical-Oriented Monitoring System (5Gr-VOMS, see §III-C). This figure explains how the typical data engineering pipeline layers have been mapped to the 5Growth architecture and provides some examples of tools for each layer. Each decision-making entity (*agent*, hereafter) in the 5Growth management platform is ultimately the one single entity that executes the model. For instance, 5Gr-SO may need a composite of neural networks to approximate the relationship between service and resource requirements [5] or to forecast demands [6]. The basic workflow for both classification/inference and reinforcement learning is the following:

0. The 5Gr-AIMLP exposes a catalog of models that can be tuned and chained to compose more complex models.
1. The agent describes the model by selecting (a composite of) preset models, their parameters for the problem at hand, as well as information on how to maintain the model and what monitoring probes are required.
2. The 5Gr-AIMLP requests 5Gr-VOMS orchestration of monitoring probes.
3. In case of reinforcement learning, the agent requests 5Gr-VOMS contextual information (e.g., current number of users) and uses it as an input of the trained model. In turn, the 5Gr-AIMLP uses such contextual information for optimization of the model parameters.
4. When the conditions for collecting data samples are met, the 5Gr-AIMLP requests and feeds the data into its fitting function to optimize the model parameters.
5. The optimized model (i.e., its parameters) is passed down to the agent for online execution by exploiting performance metrics coming from 5Gr-VOMS.

In case of reinforcement learning, the agent is also responsible for integrating on-policy (e.g., SARSA) or off-policy (e.g., Q-learning) training methods. Two specific examples leveraging on 5Gr-AIMPL are introduced in §IV.

C. *Vertical-oriented Monitoring System (5Gr-VoMS)*

The Vertical-oriented Monitoring System (5Gr-VoMS) is an extension of 5G-TRANSFORMER monitoring platform (5GT-MP), designed with the objective of supporting heterogeneous set of services and technological domains; and, likewise, novel innovations devoted to enhance end-to-end reliability (via self-healing and auto-scaling), vertical control-loops stability, and analytical features, such as forecasting, inference and anomaly detection. In order to provide such innovation support, 5GT-MP must be extended to include additional functionalities such as log aggregation, a scalable data distribution system and dynamic probe reconfiguration [7]. *Elastic stack* is included in the VoMS architecture to support log aggregation, *Kafka* distributed streaming platform as scalable data distribution system and *Elastic Beats* which will, together with *Prometheus* node exporter, assist to the dynamic reconfiguration of the monitoring probes.

Architecture: Fig. 3 describes the overall 5Gr-VoMS which includes four building blocks. The Virtual Machine, where the Monitoring Agent is installed, the *Kafka Message Queues (MQ)*, and the Monitoring Platform itself which includes the most of the components related with the monitoring.

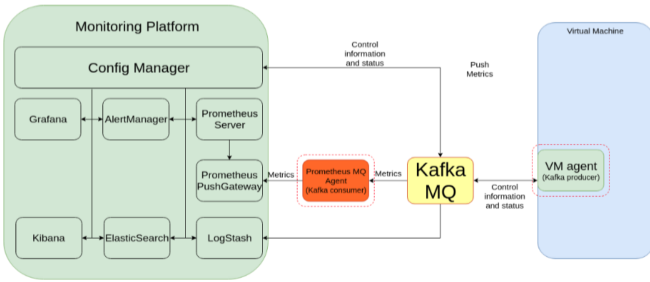


Fig. 3: 5Growth VoMS Architecture

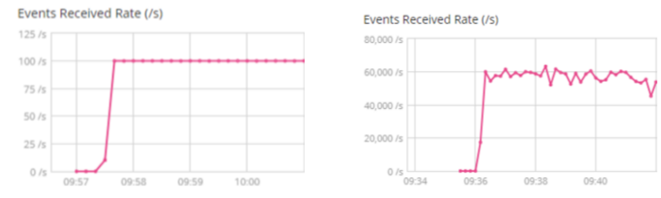
5Gr-VoMS allows using two types of time series database (TBDS) which are built specifically to handle metrics and events or measurements with time stamps. It is up to the verticals to choose which one to use, *Prometheus* or *Elastic Search* stack. *Graphana* and *Kibana* are visualization tools that allow the display and formatting of metric data obtained with *ElasticSearch* (for *Kibana*) and *Prometheus* (for *Graphana*).

The *Monitoring Agent* is responsible for collection, initial analysis and subsequent delivery of the metrics and logs in 5Gr-RL, both network and computing resources, which could be virtual or physical. There are several types of probes such as *Prometheus* exporters, *Beats* monitoring probes, etc. In Fig. 3, the *Monitoring Agent* collects the metrics and log data and pushes them to the *Kafka MQ*. On the other side, *Elastic Search* is reading the *MQ* using *Logstash*. The *Prometheus MQ agent* acts as an intermediary between *Kafka* and *Prometheus*. Logs and metrics are extracted and placed in the TSDB once they appear in the MQ.

MQs are used as an interface for information exchange between different technologies and components of the architecture. In this way, internal and external components (e.g., federated domains, etc.) can read/publish information in a common way, avoiding the definition, creation, and implementation of new APIs. If needed, creating new *MQs* and add them to the stack is straightforward and does not increase the complexity of the architecture. Fig. 3 shows the case of Virtual Machines. The *Config Manager* configures the Monitoring Agent. Finally, for integration purposes, *Prometheus* and *Elastic Search* have an API to provide information to other modules such as Anomaly Detection, Forecasting and Inference or *Alert Manager*.

Preliminary Results: This subsection describes a set of experiments that have been performed with the purpose of validating the VoMS innovations. Specifically, they target the evaluation of the **scalability** of the main component introduced to the architecture, the *Kafka* message queue. The experiments are performed instantiating the different components of the 5G-VoMS in a *Docker* container. Furthermore, an external Virtual Machine containing the *Berserker* tool is connected to the VoMS through the *Kafka* message queue. This tool allows to generate monitoring information messages at variable rates, which in this case is used to emulate monitoring probes. The hardware equipment is provided with 8 CPU cores, 8GB RAM and 100GB of disk. The *Kafka* JVM heap memory is 4GB.

Fig. 4 shows the number of events received from *Kafka*'s MQ by *Logstash*, when the Berseker tool is configured to generate monitoring load at a rate of 10^2 and 10^5 messages/s, respectively. The graphs are obtained using the *Kibana* visualization tool from the VoMS. In Fig. 4a, it can be observed that the *Kafka* component is able to maintain the rate of 100 messages processed per second. On the other hand, in Fig. 4b it can be appreciated that, when the number of messages generated is 100000 messages per second, the maximum number of messages that *Kafka* is able to process oscillates around 60000 messages/s. This result demonstrates the **high scalability** of the *Kafka* message queue, given that this scenario would be equivalent to a scenario where 60000 probes are publishing monitoring information at a pace of one message per second.



(a) Load equal to 10^2 messages/s (b) Load equal to 10^5 messages/s

Fig. 4: Number of *Kafka* MQ events as received by *Logstash*.

Fig. 5: Latency associated to each *Logstash* event.

Furthermore, Fig. 5 shows the latency of each event processed by *Logstash* when the load is equal to 10^5 messages/s, which is approximately constant at around 0.18 ms. From this result, it can be concluded that even though the *Kafka* MQ has reached its performance saturation point, the rest of components of the architecture that process the events generated by *Kafka* (in this case, *Logstash*), do not experiment performance degradation, validating the platform scalability.

IV. Smart Orchestration and Control

Deployment of the requested NFV-NS is a two-step process using the 5Growth software stack. Each step bases its decisions on its own specific abstraction of the information provided for the underlying infrastructure(s).

1) 5Gr-SO decides upon the optimal NFV-NS decomposition based on service availability as well as resource capabilities exposed at the local and other administrative domains. Towards that end the 5Gr-SO builds up an abstract view (i.e. annotated topology) of the federated infrastructure, by exchanging abstract views (e.g., abstract topologies, computing and storage capabilities) with other domains and consolidating them with the local view exposed by the resource layer. This is used for optimal placement of network functions at the set of interconnected NFVI PoPs and the optimal deployment of virtual links connecting them. The process amounts to NFV-NS decomposition, as essentially different service segments of the initial NFV-NS graph are mapped to different PoPs.

2) 5Gr-RL receives the aforementioned service segments from the 5Gr-SO and bindings for their corresponding end-points. These segments are expressed as VNF-Forwarding Graphs. 5Gr-RL is responsible for mapping the requested resources to the underlying infrastructure, thus addressing the corresponding resource allocation problem known as VNF-FG embedding [8]. VNF-FG embedding involves placement of its constituent VNFs at the set of interconnected physical nodes within the managed domain and in-sequence routing through them as prescribed by the forwarding graph. In the following we will present our initial attempt to address the corresponding problem using ML in the context of 5Growth.

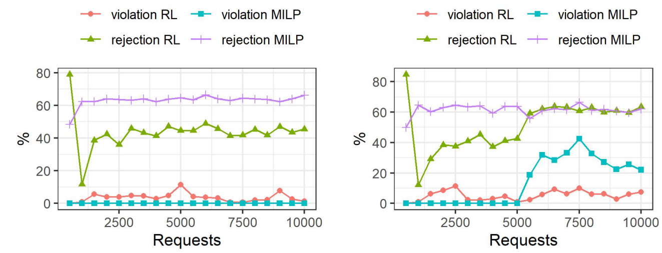
Towards intelligent resource allocation for 5G and beyond, Dynamic Profiling Mechanims will be used to extract resource demands for the underlying network components. The resource demands will be eventually used as input for the 5Growth network optimization solutions (i.e., pertaining to resource allocation and scheduling).

A. VN-FG Embedding

The VNF-FG embedding problem is often formulated as a Mixed Integer Linear Program (MILP), tailored to the specific objective that is pursued e.g., [9][10]. The solution determines the placement of the VNF-FG nodes on the servers and the mapping of the directed VNF-FG edges on substrate paths. Since the problem is NP-hard [8] sub-optimal (meta) heuristics and approximation algorithms have been devised to make the problem computationally tractable, considering that mapping needs to be addressed in real time ("online") problem.

Most approaches dealing with the online problem make decisions based on a snapshot of the residual capacities in the NFVI observed at request time, and it is usually assumed that these capacities are known with high precision, while the (future) evolution of the workloads for in-service (or expiring) VNF-FGs over time is not considered. The former assumption is unrealistic given the coarse-granularity of the monitoring information in time, e.g., to keep the corresponding network overhead low. Moreover, making embedding decisions based only on a snapshot of the remaining resources at request time is not optimal over time, as it leads to fragmentation of the physical resources. What is more, the maximum (or average) of resources that an VNF-FG may require over its lifetime is considered, which leads either to over-provisioning of resources (hence under-utilizing the physical infrastructure and rejecting incoming requests) or SLA violations.

In such an environment with uncertainty in resource demands and provisioning, reinforcement learning is suited to tackle the VNF-FG embedding problem. The reinforcement learning based approach gradually steers the decision-making process in the right direction based on feedback it gets on how good the embedding decisions were. Concretely, at the end of each episode (of e.g., 500 requests), the 5Gr-AIMLP platform will be called upon to adapt the policy based on the (state, action, reward) triplets that were observed over that episode. The approach can be used to address the "online" problem, supporting decision-making in real (polynomial) time. Each time a VNF-FG arrives at the 5Gr-RL, the reinforcement learning agent decides if (admission control) and how (mapping) to embed the VNF-FG in the NFVI. All constraints imposed by the problem at hand (related to capacity, QoS, etc.) are translated into rewards (made up of bonuses and penalties); by rewarding actions that accept the requested VNF-FG and do not violate constraints while penalizing the ones that do, the ML-based algorithm gradually learns the best policy.



(a) Scenario 1 (b) Scenario 2

Fig. 6: Resource violation - request rejection.

Preliminary Results: We compare the efficiency of the reinforcement learning approach, denoted as **ML**, to the benchmark baseline **MILP** [11] using simulations. The MILP uses as traffic envelope the maximum inbound traffic demand per service chain. We compare them on the basis of (i) the VNF-FG request rejection ratio defined as the ratio of rejected requests divided by the total number of requests, and (ii) the resource violation ratio defined as the ratio of the monitoring instances at which any of the resources is violated to the total number of monitoring instances (thus implicitly considering SLA violations). We use an event-based simulator implemented in Java, including an SFC and DataCenter (DC) topology generator. The ND4J (see https://deeplearning4j.org/) library has been adopted for tensor operations support. We use CPLEX (branch-and-cut) for our MILP models.

Indicatively, two simulation scenarios are evaluated. For the first simulation scenario, we compare the efficiency of the reinforcement learning approach. Fig. 6a depicts the evolution of the two metrics for the different approaches. The ML-based approach converges after approximately 1500 requests. In steady state there are still fluctuations due to the stochastic nature of the requests and the exploration capability of the RL approach. MILP has no violations by design but exhibits the highest rejection ratio as it takes into account the maximum inbound traffic demand per service chain. The ML-based approach manages to keep the resource violation ratio low, without considering capacity constraints for the embedding problem and having limited information on the infrastructure resources, as opposed to the MILP that is provided with the remaining compute and transport capacity in full precision.

For the second scenario, we study the ability of the reinforcement learning-based approach to adapt fast to changing conditions such as a surge in workload/traffic demands. To assess this aspect, we increase the requested workload halfway through the simulation; for the 5000 remaining VNF-FG requests the corresponding inbound traffic is increased approximately by 30%. Fig. 6b shows that the proposed ML-based approach adapts to this new situation by rejecting more requests keeping the violation ratio more or less constant, because the rewards were set such that violations are expensive and rejections are rather cheap. Convergence to the new "steady state" is fast. The MILP approach is not able to cope with these changing conditions: it has much more violations while it was designed to avoid those in the first place. After the load increase, the resource violations in the MILP case could only be avoided by resetting the traffic envelope for the incoming requests at the second half of the simulation.

B. Dynamic Profiling Mechanism

Dynamic Profiling Mechanism (DPM) builds upon the 5Gr-AIMLP introduced in §III-B to extract network behavior- and service usage-based UE profiles. The DPM, which extends the functionality of the Context Extraction and Profiling Engine (CEPE) [12], extracts a set of UE profiles, based on past behavior in terms of UE capabilities, mobility patterns and resource requirements and forwards them to the resource allocation and smart orchestration layers of the NFV-RO of 5Gr-SO and the 5Gr-RL.

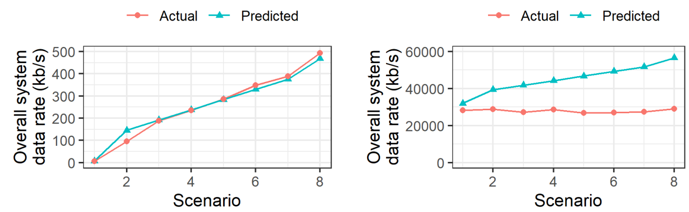
The goal of DPM is to extract UE profiles based on UE- (user and device), network-, service- and slice-oriented contextual information, following a step-by-step methodology.

1) Data Management/Collection: Collection of data from multiple sources based on 5Gr-AIMLP requests to 5Gr-VOMS, cleaning, filtering, and correlation of data;
2) Application of Divisive Hierarchical Clustering models and fine-tuning inside the 5Gr-AIMLP platform, in order to construct classes with similar observations;
3) Application of a predefined set of rules in conjunction with Decision Tree Learning Algorithm 5Gr-AIMLP models in order to extract the necessary profiles; and
4) Extraction of profiles and forwarding towards the respective agents for RAN resource allocation, VNF autoscaling and placement schemes in 5Gr-SO and 5Gr-RL.

The input data comprises diverse datasets collected from different parts of the network and relating to user data, device information, service levels and network resources.

We next present an initial evaluation on a RAN resource allocation approach, where we focus on a single eMBB slice for simplicity. Our approach extracts a first set of profiles based on the past behavior of UEs in terms of network service type they consume. The evaluation is done with NS-3 network simulator. In this initial simplified evaluation, 5 different UE profiles were used, consuming different network services with different UL/DL data rate requirements. Overall, 8 scenarios were executed involving 40 UEs with different profile distribution probabilities. The Profiling Mechanism classified the UEs into service profiles correctly, which allows us to proactively allocate the respective resources.

The results, presented in Fig. 7 compare the predicted and actual resources that were finally used during the UEs' activity in the uplink and the downlink channels. Although the prediction accuracy in the UL case is clearly higher, in both cases the predicted resources allocated were equal or more than the ones finally used.



(a) Uplink (b) Downlink

Fig. 7: Predicted *vs.* actual resource consumption during the UEs activity in both uplink and downlink channels.

V. Conclusion

This paper introduced some of the innovations proposed by the H2020 5Growth project. Specifically, we have presented initial work and results regarding (i) architectural innovations to apply novel AI/ML schemes into management operations, (ii) vertical control over radio resources, (iii) enhanced monitoring service, and (iv) automated service orchestration mechanisms. These initial results (among others that will be integrated in the future) make evident how 5G paves the way to innovative use cases in vertical industries and novel service management procedures. The project is currently on its first year, with initial pilots under design, involving verticals alongside the development of the identified innovations, with first field trials projected to the end of 2020.

Acknowledgments

This work has been partially supported by EC H2020 5GPPP 5Growth project (Grant 856709).

References

[1] A. De la Oliva *et al.*, "5G-TRANSFORMER: Slicing and orchestrating transport networks for industry verticals," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 78–84, 2018.
[2] 3GPP, "Technical Specification Group Services and System Aspects; Management and orchestration; 5G Network Resource Model (NRM); Stage 2 and stage 3 (Release 16)," 3rd Generation Partnership Project (3GPP), TS 28.541, September 2019.
[3] ——, "Technical Specification Group Services and System Aspects; System Architecture for the 5G System (Release 16)," 3rd Generation Partnership Project (3GPP), TS 23.501, March 2019.
[4] D. M. Gutierrez-Estevez *et al.*, "Artificial intelligence for elastic management and orchestration of 5g networks," *IEEE Wireless Communications*, vol. 26, no. 5, pp. 134–141, 2019.
[5] J. A. Ayala-Romero *et al.*, "vrAIn: A Deep Learning Approach Tailoring Computing and Radio Resources in Virtualized RANs," in *Proc. of ACM MobiCom 2019*.
[6] J. X. Salvat *et al.*, "Overbooking network slices through yield-driven end-to-end orchestration," in *Proc. of ACM CoNEXT 2018*.
[7] Y. Wei, M. Li, and B. Xu, "Research on establish an efficient log analysis system with kafka and elastic search," *Journal of Software Engineering and Applications*, vol. 10, no. 11, pp. 843–853, 2017.
[8] J. G. Herrera and J. F. Botero, "Resource allocation in NFV: A comprehensive survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 518–532, 2016.
[9] D. Dietrich *et al.*, "Network function placement on virtualized cellular cores," in *Proc. of COMSNETS 2017*, pp. 259–266.
[10] N. Torkzaban, C. Papagianni, and J. S. Baras, "Trust-aware service chain embedding," in *2019 Sixth International Conference on Software Defined Systems (SDS)*, June 2019, pp. 242–247.
[11] C. Papagianni, P. Papadimitriou, and J. S. Baras, "Rethinking Service Chain Embedding for Cellular Network Slicing," in *2018 IFIP Networking Conference (IFIP Networking) and Workshops*, May 2018, pp. 1–9.
[12] P. Magdalinos *et al.*, "A context extraction and profiling engine for 5G network resource mapping," *Computer Communications*, vol. 109, pp. 184–201, sep 2017.