# Data-Driven Estimation of Throughput Performance in Sliced Radio Access Networks via Supervised Learning

C. Gijón, M. Toril, S. Luna-Ramírez
Telecommunication Research Institute (TELMA), Universidad de Málaga, Málaga, Spain.

***Abstract*—In 5G systems, Network Slicing (NS) feature allows to deploy several logical networks customized for specific verticals over a common physical infrastructure. To make the most of this feature, cellular operators need models reflecting cell and slice performance for re-dimensioning the Radio Access Network (RAN). For enhanced Mobility BroadBand (eMBB) services, throughput is regarded as a key performance metric since it strongly influences user experience. This work presents the first comprehensive analysis tackling cell and slice throughput estimation in the downlink of RAN-sliced networks through Supervised Learning (SL), based on information collected in the operations support system. Different well-known SL algorithms are tested in two NS scenarios with single-service or multi-service slices serving eMBB users. To this end, several synthetic datasets are generated with a system-level simulator emulating the activity of a sliced RAN. Results show that NS alters the correlation between network performance indicators and cell throughput compared to legacy RANs, thus being required a separate analysis for NS scenarios. Moreover, the best model to estimate throughput at cell/slice level may depend on the scenario (single-service vs multi-service slices). In all cases, the best models have shown an estimation error below 10 %.**

***Index Terms*—Network slicing, radio access network, supervised learning, throughput, enhanced mobility broadband.**

## I. Introduction

5G technology has been conceived to expand the business model of cellular operators by providing vertical industries with enhanced Mobile BroadBand (eMBB), ultra-Reliable Low Latency Communications (URLLC) and massive Machine Type Communications (mMTC) services [1]. Such a trend will continue in beyond 5G and 6G networks offering new services, such as holographic teleoperation, internet-of-everything or extended reality [2]. To fulfill stringent 5G and 6G requirements, features such as network virtualization, cloudification, edge computing or Network Slicing (NS) have been proposed [3]. The latter is considered a very promising solution to guarantee the highly diverging Quality of Service (QoS) requirements of different services (e.g., large bandwidth for eMBB or reduced latency for uRLLC) [4].

NS consists in deploying multiple logical networks on a common physical network infrastructure [5]. This functionality provides the flexibility and scalability required to accommodate the diverse devices, use cases and service types co-existing in 5G networks. Slices are self-contained and logically isolated, which increases robustness (i.e., faults in one slice have no effect on other slices) and security (i.e., an attack to a slice will not affect other slices) and reduces time-to-market (i.e., there are few dependencies on external network functions –NFs–). Moreover, slice NFs and resources are tailored to meet the requirements of a specific type of business or service in terms of performance, security, mobility and availability. As a consequence, NS maximizes revenues for infrastructure owners due to the efficient usage of network assets while opening up new go-to-market models for verticals [6].

An end-to-end slice comprises a collection of NFs and hardware, software and radio resources customized for a specific business model associated with a particular application and/or tenant (e.g., on-the-top service providers or virtual operators) [5]. In the Radio Access Network (RAN), NS entails implementing new Radio Resource Management (RRM) NFs to efficiently create, (de)activate and operate slices whilst guaranteeing slice isolation (e.g., radio capacity brokers or slice admission and congestion control policies). To support RRM NFs, radio network performance indicators must be monitored on a slice basis to guarantee the fulfillment of Service Level Agreements (SLAs) and operate the business associated to each slice [7].

The slice life cycle comprises the preparation, commissioning, operation and decommissioning phases [8]. In the preparation phase, the SLA is defined. Then, in the commissioning phase, the capacity conformance NF estimates the amount of resources to be assigned to the slice per cell in order to fulfill performance requirements in the SLA while maximizing network profits [9]. For this purpose, in the RAN, slice-level models are used to map specific slice characteristics (e.g., traffic type and demand, bandwidth, spectral efficiency...) to performance metrics. These slice-level models are also continuously exploited during the slice operation phase to check when spectrum sharing has to be reconfigured to meet the SLA while minimizing resource over-provisioning [10]. Thus, the development of accurate slice-level performance models has gained interest for Mobile network Operators (MNOs). For slices serving eMBB traffic, throughput is often the most highly-demanding performance requirement among those included in SLAs (e.g., end-to-end latency, reliability. . . [11]), while having a large impact on user experience.

In parallel, NS poses additional challenges to infrastructure owners when re-planning their RAN. Radio planning tools compare traffic forecasts with estimated network capacity to detect bottlenecks in advance and execute re-planning actions if necessary [12]. For this purpose, cell-level performance models are used to estimate cell capacity under the expected future conditions (i.e., traffic demand, radio channel quality, cell configuration…). Cell capacity is usually measured as aggregated cell throughput in high load conditions [13] [14]. However, performance models used in legacy networks may fail in a NS scenario due to: a) the coexistence in the same geographical area of multiple slices with very different behaviors over the spatial and temporal domains, and b) the split of radio resources among slices, which may prevent the efficient use of cell bandwidth. Thus, it is necessary to check if cell-level performance models used in non-NS networks are still valid in RAN-sliced scenarios.

With the latest advances in information technology, it is possible to create performance models with Big Data Analytics (BDA) techniques that take advantage of massive data collected in the Operations Support System (OSS). The most advanced data-driven management tools rely on Machine Learning (ML) techniques, considered key for managing the upcoming cellular networks [15]. This ML-based approach can be used to estimate slice and cell performance in complex RAN-sliced networks, since it allows to create models adapted to each particular scenario by considering the specific RRM algorithms (e.g., capacity broker), NS framework (e.g., single-service slices or multi-service slices) and service mix (e.g., cloud gaming, web browsing, autonomous driving…) [16].

In this work, we present the first comprehensive analysis to compare the performance of several well-known SL algorithms to estimate cell and slice throughput in the Down Link (DL) in a RAN-sliced network from information collected in the OSS. The analysis focuses on eMBB services, for which throughput is key to guarantee user satisfaction. The SL approach allows to capture the impact of a specific network configuration (i.e., NS set-up, capacity broker…) on DL throughput, which is critical for an effective network re-dimensioning in live RAN-sliced networks (e.g., for predicting cell capacity bottlenecks in the presence of slices and re-configuring resources for a particular slice). For this purpose, synthetic measurement datasets have been created with a dynamic system-level simulator emulating the activity of a real cellular network.

The rest of the document is organized as follows. Section II revises related work. Section III outlines the problem of estimating radio throughput performance from statistical measurements in RAN-sliced scenarios. Section IV describes the dynamic system-level simulator used as a system model. Section V details the considered methodology for estimating throughput at a cell and slice level. Section VI presents model assessment. Finally, section VII summarizes the main conclusions.

## II. Related work

In this section, the literature related to this work if first reviewed, and then the motivation and contributions are exposed.

### A. State of the art

In the literature, several radio throughput performance models tuned with real network statistics have been proposed for re-planning purposes. In [17], an analytical performance model of dynamic packet scheduling in Long Term Evolution (LTE) is presented to estimate radio user throughput in a multi-service scenario. Model parameters are adjusted with information from radio connection traces to reflect a particular scenario. Alternatively, a simpler approach is to estimate cell performance with regression models based on data collected in the OSS. Preliminary works relied on Multiple Linear Regression (MLR). In [18], a performance model based on MLR is derived to estimate DL cell throughput in the busy hour in High Speed Downlink Packet Access (HSDPA) from network performance counters and configuration settings collected on a cell basis. In [19], this approach is extended to estimate the same metric in a LTE network from propagation, channel quality and delay information for a specific cell. In [13], it is shown with real data that MLR can estimate cell throughput reasonably well in a multi-service LTE network, but not packet delay statistics of Voice over Internet Protocol (VoIP) users. Recent works use more complex SL methods to capture non-linear dependencies among input and output features. In [14], different SL algorithms are used to estimate cell and average user throughput in HSDPA and LTE DL. The most relevant features per algorithm, radio access technology and metric are identified through a sequential feature selection process, resulting in simplified models with estimation errors below 10 %.

In 5G systems, assessing network performance becomes more complicated with NS. End-to-end (E2E) NS entails deploying several logical networks over a common physical infrastructure in the core, transport and radio access domains, managed by a Management and Orchestration (MANO) system. In [20], different architectures to provide E2E NS relying on software-defined networks, NF virtualization and cloud/fog computing are surveyed. The information model required for NS in RAN, core and transport network domains is described in [21]. Other works focus on MANO, defining slice management policies for tasks such as slice admission control [22] or resource allocation [23]. In RAN, NS management is more challenging due to the inherently limited and shared nature of spectrum. In [24], four different strategies to split radio resources among slices are compared in terms of spatial, temporal and frequency granularity of assignment, traffic and radio isolation, and customization. Several capacity brokers have been proposed following these strategies, sometimes joining spectrum split and access control tasks [25].

Due to the complexity of RAN slicing mechanisms, cellular operators need new performance models at cell and slice level for (re)dimensioning their networks. In [26], an analytical model is presented to estimate user blocking probability in a cell serving guaranteed-bit-rate slices from channel quality information. The model is based on a multi-dimensional Erlang-B system, insensitive to session duration distribution. An analytical approach is also considered in [27] to estimate the required capacity per slice on a cell and pixel basis for re-dimensioning purposes. The model is fed with cell configuration, channel quality information, and traffic density and volume per active slice.

In parallel, BDA has been identified as a key enabler for automatic network management to cope with the complexity, diversity and dynamicity of current 5G (and future 6G) systems [2]. Examples of network management issues solved by BDA are capacity and coverage optimization [28], mobility load balancing [29], network dimensioning [30] or alarm prioritization [31]. A comprehensive survey of ML methods used in self-organizing cellular networks can be found in [32]. In future 6G systems, ML will be integrated in many different network entities to allow advanced radio interfaces, optimal traffic control, intelligent orchestration and reliable network security [33].

In NS scenarios, ML-based solutions have been proposed for resource split among slices [25] [34], slice admission control [25], user-centric slice design [35] or slice classification per service type [36], among other tasks. Closer to the work presented here, in [37], SL is used to estimate application-level video requirements from low-layer network measurements to improve the slice negotiation phase. In [38], a digital twin network model relying on graph neural networks is used to predict end-to-end packet latency in three different NS scenarios, capturing intertwined relationships among slices.

### B. Motivation and contributions

Previous works like [19] [29] state the interest of mobile network operators on SL-based cell-level throughput estimation models. As explained in the related work section, contributions in the literature focus on legacy (i.e., non-sliced) networks. However, slicing the RAN implies significant changes in the network (e.g., a slice can only use a subset of radio resources) that, as shown later in this work, alter the correlation between network indicators and throughput. Thus, cell throughput estimation in NS scenarios requires a separate analysis. Moreover, new network functions arise when enabling NS (e.g., capacity brokers), which require slice-level throughput estimation models for assigning spectrum to slices guaranteeing SLA fulfillment and an efficient use of system bandwidth. To the authors' knowledge, the performance of SL models to estimate cell or slice throughput in RAN-sliced networks has not been assessed yet. These problems are tackled in this work.

Network operators are reluctant to use complex deep learning models with thousands of hyperparameters in their network management tools, since these models are difficult to interpret and prone to overfitting if not trained with very large datasets (i.e., hundreds of thousands of samples). Under this premise, this work focuses on checking if simpler classical SL algorithms suffice to solve the considered problem. All SL and feature selection techniques considered here are well-established schemes included in most data analytics packages. Hence, the main novelty here is the assessment of these classical methods for the new problem of estimating cell and slice throughput in sliced RANs. Specifically, the main contributions of this work are:

- Presenting the first study assessing the performance of well-known SL models to estimate DL slice throughput in RAN-sliced networks from information in the OSS. The considered approaches include classical neural networks, ensemble models, distance-based models and vector-based models that have never been used for slice performance estimation.
- Assessing the performance of these algorithms to estimate DL cell throughput in RAN-sliced networks. We added two ensemble models (adaptive boosting and extreme gradient boosting) to the algorithms tested in [14] in legacy scenarios. To justify the need for this contribution, we present an analysis of the impact of enabling NS on the correlation between network indicators and throughput.
- Extending the comparative analysis to two different NS scenarios, comprising single-service and multi-service slices serving eMBB users.
- Identifying a minimal set of key network performance indicators to be stored in the OSS to estimate throughput in RAN-sliced networks. The novelty here is the inclusion of features derived from radio connection traces. This source of data was not considered in [14], where input features only considered performance management (PM) and configuration management (CM) data.

## III. Problem outline

The aim of this work is to develop models to estimate cell/slice throughput in the DL at a given time $t$ from information (real or hypothetical) on network state at time t. These models are key for efficient network re-dimensioning, since they allow: a) to analyze a worst-case scenario for the current network set-up, and b) to assess the impact of re-dimensioning actions (e.g., cell/slice bandwidth extension/reduction, deployment/temporal switch-off of a cell…) on cell and slice performance. To capture network peculiarities, models are built from information collected in the OSS.

The estimation of DL throughput of an entity $k$ (i.e., cell or slice) of a cellular network in the RAN, $TH(k)$, from information collected in the OSS can be tackled as a regression problem. Throughput depends on many factors related to radio channel conditions (e.g., indoor/outdoor environment, inter-site distance...), network configuration (e.g., packet scheduling algorithm, radio resource utilization threshold...) and user profile (e.g., traffic mix, terminal capabilities...). As a consequence, complex regression models with dozens of input features (predictors) can be derived. In the simplest model, DL throughput estimation problem is formulated as

$$\widehat{TH}(k) = f(C(k), \%C_{util}(k), SE(k)) \ . \tag{1}$$

where $C(k)$ denotes capacity of entity $k$, $\%C_{util}(k)$ is the amount of used capacity, and $SE(k)$ is spectral efficiency, reflecting how much data can be transmitted per capacity unit with radio link conditions experienced in the DL of entity $k$.

The activation of NS functionality can have a strong impact on network performance. Cell bandwidth, Physical Resource Block (PRB) utilization ratio in the Physical Downlink Shared Channel (PDSCH) and average Channel Quality Indicator (CQI) are often considered as capacity, used capacity and spectral efficiency indicators, respectively, when estimating DL cell throughout. In live networks, cell bandwidth determines the maximum achievable cell throughput, no matter if NS is enabled or not. In legacy networks without NS, all users share the spectrum, leading to a high PRB utilization ratio in peaks periods in the presence of users demanding data-hungry services. As a consequence, for a specific cell bandwidth, DL cell throughput strongly depends on spectral efficiency (i.e., CQI reported by UEs), which determines the amount of bits that can be transmitted per PRB. In contrast, in NS scenarios, the split of radio resources among slices may prevent the packet scheduler to make the most of cell bandwidth. If this is the case (as will be shown later in this work), the PRB utilization ratio becomes a relevant indicator to estimate DL cell throughput. Such differences suggest that new cell-level performance models must be derived for NS scenarios.

Apart from cell-level models, in RAN-sliced networks, some advanced RRM tasks require estimating system performance with a finer resolution. For instance, slice management policies guaranteeing SLA fulfillment need a model to estimate DL throughput per slice. For this purpose, the number of PRBs per slice may be used as a slice capacity indicator in (1), whereas PRB utilization and CQI statistics per slice can be used as metrics of used capacity and spectral efficiency. Note that slices can be set to serve only a type of service (e.g., videostreaming slice) or a mix of services (e.g., all traffic belonging to a virtual MNO). To make it easier for operators to select the best slice set-up, it is convenient to study the impact of service mix when estimating slice performance by comparing results from scenarios with single-service and multi-service slices.

In 5G systems, the increase of bursty data from services with small packet size (e.g., mMTC services) may alter the correlation between network indicators (e.g., number of simultaneous users) and DL slice throughput. Previous cell performance models have been developed using data from live 3G or 4G networks, where most connections belong to data-hungry services [39]. To deal with service diversity, it could be necessary to include features reflecting the traffic mix in network performance models at cell and slice level.

Note that, in the RAN, several important aspects affecting slice definition are up to MNOs. For instance, depending on the selected slicing approach, radio resource split among slices might be flexible (i.e., slices share resources) or dynamic (e.g., on a minute or slot basis), resulting in different radio-electrical and traffic isolation between slices [24]. Likewise, RRM algorithms (e.g., packet scheduling or access control) can be customized per tenant, leading to different slice behavior. Thus, when estimating cell or slice performance, it is convenient to derive empirical models based on SL algorithms to capture non-linear relationships among features and peculiarities of each specific NS scenario (i.e., capacity broker function, slices offering one service or a mix of services...).

## IV. System model

Since large-scale datasets from operational networks with NS are not available yet, datasets used in this work have been created with a dynamic system-level simulator emulating the activity of a real cellular network. In this section, the main characteristics of the simulator are first described. Then, the considered NS implementation is detailed. Finally, the different NS scenarios configured in the simulator are introduced.

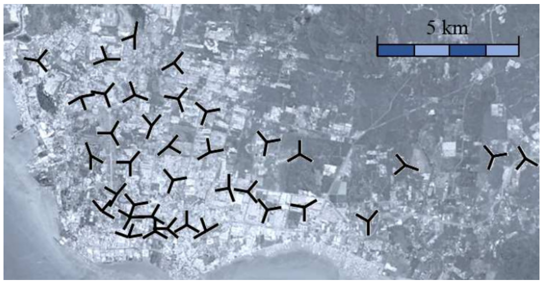

Fig. 1: Simulated scenario [40].

TABLE I: Simulation parameters.

| Parameter | Description |
|---|---|
| Time resolution | 10 ms |
| Carrier frequency | 2.1 GHz |
| Transmission mode | FDD |
| System bandwidth | 5 MHz / 10 MHz |
| 5G numerology ($\mu$) | 0 |
| Propagation model | Path loss: Hata COST-231<br>Slow-fading: log-normal $\sigma$ = 8 dB, $d_c$ = 40 m<br>Fast fading: ETU model |
| Base station model | Tri-sectorized antennas, MIMO 2x2, transmit power 49 dBm, no beamforming |
| Packet scheduler | Classical exponential/proportional fair [44] |
| Link adaptation | CQI-based, MCS selected to guarantee BLER$\leq$0.1 |
| Traffic model | Non-uniform spatial UE distribution and traffic mix<br>Ratio of UEs per service in a cell: uniform [0.02, 0.7] |
| VoIP model | Coding rate: 16 kbps, duration: exponential (avg. 60 s), call dropped after 1 s without resources |
| Video model | Packet arrival process and file size from H.264/MPEG-4 AVC real traces, resolution: 720p, duration: uniform [30,540] s |
| File download model | FTP, file size: log-normal (avg. 10 MB) |
| Web browsing model | HTTP, no. pages per session: log-normal (avg. 4), reading time: exponential (avg. 30 s) |
| UE speed | 3 km/h (30 % of UEs), 0 km/h (70 % of UEs) |
| Handover (HO) set-up | Power budget HO. HO margin of 3 dB and time to trigger of 512 ms for every adjacency |

### A. General description

NS functionality has been integrated into an existing LTE-Advanced simulator that emulates the activity of the RAN in a live network [29] [40]. Whenever possible, simulation set-up follows New Radio (NR) specifications. The simulated scenario, illustrated in Fig. 1, consists of 108 irregular cells located in urban and sub-urban areas covering 11×23 km$^2$. Table I breaks down the main simulation parameters. Cells work at 2.1 GHz. As specified in NR standards for this frequency band, the network operates in Frequency Division Duplexing (FDD) mode [41]. Different bandwidths in [41] are considered in different simulations, as explained later. A subcarrier spacing of 15 kHz (i.e., 5G numerology with $\mu$ = 0) is selected. Modulation and Coding Schemes (MCS) used are those in the 4-bits CQI table in [42]. A link abstraction model maps Signal to Noise and Interference Ratio (SINR) to Block Error Rate (BLER) on a certain MCS [43]. For each User Equipment (UE), the more efficient MCS guaranteeing BLER$\leq$0.1 is selected. For computational efficiency, only the DL is simulated with a time resolution of 10 ms

User distribution in the scenario is non-uniform and based on that from the live network. In a cell $c$, new UEs appear following a Poisson process with $\lambda(c)$. The generated UEs demand 4 services: VoIP, progressive video streaming, file download via File Transfer Protocol (FTP) and web browsing via HyperText Transfer Protocol (HTTP). Whilst VoIP is a delay-sensitive service with low code rate, the remaining services are data-hungry and mainly throughput sensitive (i.e., as eMBB). Table I includes the main service parameters regarding traffic model and connection duration. As shown in the table, service mix also varies per cell. Regarding UE speed, 70 % of UEs are static (e.g., indoor users), whereas the remaining 30 % are pedestrians. For more details of service implementation and propagation model in the simulator, the reader is referred to [40].

### B. Network slicing implementation

As in [27], the SLA is defined in terms of capacity requirements for the expected traffic in a given area. In the preparation phase, a tenant applying for a slice $i$ provides the operator an Individual Slice Template (IST), including a) the slice operation area, b) the expected service spatial distribution and traffic intensity (i.e., UEs) in peak periods and c) the required average session throughput per service $s$ offered in the slice, $TH_{APP}(i,s)$, defined at application layer. For VoIP service, $TH_{APP}(i,VoIP)$=16 kbps (i.e., the coding rate). For the remaining services, target throughput is set by the tenant to guarantee a good QoE for all services according to Mean Opinion Score (MOS) models in [29] (i.e., MOS between 4 and 5).

In the planning phase, the capacity conformance NF uses an analytical model to determine the spectrum allocation needed to fulfill the SLA. The number of PRBs required by slice $i$ in cell $c$, $N_{PRB}(c,i)$, is the aggregation of PRBs required to schedule UEs demanding all services $s$ offered in the slice, $N_{serv}(i)$,

$$N_{PRB}(c,i) = \sum_{s=1}^{N_{serv}(i)} N_{PRB}(c,i,s) \ . \tag{2}$$

For a given service $s$, $N_{PRB}(c,i,s)$ can be estimated as

$$\widehat{N_{PRB}}(c,i,s) = N_{UE}(c,i,s)\, \widehat{N_{PRB\_UE}}(c,i,s) \ , \tag{3}$$

where $N_{UE}(c,i,s)$ is the expected average number of simultaneous RRC connected UEs from slice $i$ demanding service $s$ in cell $c$ and $\widehat{N_{PRB\_UE}}(c,i,s)$ is the estimated number of PRBs required by each individual UE in cell $c$ to fulfill service requirements in the SLA. The former term can be computed straightforward from traffic information in the SLA, whereas the latter term can be estimated as

$$\widehat{N_{PRB\_UE}}(c,i,s) = \frac{TH_{PHY}(i,s)}{\overline{TH_{PRB}}(c)} \ , \tag{4}$$

where $TH_{PHY}(i,s)$ is the throughput required at physical layer to achieve the target $TH_{APP}(i,s)$ and $\overline{TH_{PRB}}(c)$ is the DL throughput per PRB experienced by an UE with average SINR in the service area of cell $c$. For VoIP (i.e., small data chunks), headers are considered to compute $TH_{PHY}(i,s)$. In contrast, for the remaining considered services, header size is negligible compared to data chunk size, and thus it is assumed that $TH_{PHY}(i,s) \approx TH_{APP}(i,s)$. To avoid underestimating required capacity in cells with low traffic, a minimum value of $N_{PRB}(c,i)$=3 is set for every cell and slice.

In the operation phase, as in [34], the capacity broker NF periodically adjusts $N_{PRB}(c,i)$ per cell and active slice by re-assigning underutilized PRBs to slices whose capacity requirements have been underestimated. This process is repeated every 5 minutes until a steady state is reached. To make the most of spectrum capacity, the minimum slice chunk is reduced to 2 PRBs.

The above-described spectrum sharing scheme is well aligned with previous proposals in the literature [24]. Intra-cell traffic isolation is ensured, since the packet scheduling function of a slice can only use PRBs assigned to that slice, thus preventing a high-load period in a slice from affecting other slices. However, since PRB assignment may be different in adjacent cells, inter-cell traffic isolation is not guaranteed. Nonetheless, spectrum splitting is performed by minimizing the probability of assigning a certain PRB $p$ to different slices in neighbor cells so as to reduce inter-cell inter-slice interference.

### C. Scenarios

The following three scenarios are considered:

- Scenario with NS and single-service slices (NS_SS): in this scenario, all cells in the network allocate 4 slices, which remain active for the whole simulation. Each slice exclusively offers a single service (i.e., VoIP, video, file download or web browsing). This scenario is representative of a system where the MNO creates slices optimized to fulfill certain service requirements of specific clients or those of OTT service providers.
- Scenario with NS and multi-service slices (NS_MS): in this scenario, there are also 4 slices whose operation areas cover the whole network. However, unlike the previous case, all slices offer all services, emulating a network with 4 virtual MNOs operating on different slices over the same infrastructure.
- Scenario without NS (noNS): a legacy network scenario, where all UEs share the available bandwidth.

## V. Throughput estimation method

This section explains the methodology used to estimate cell and slice throughput. The process consists of four steps. First, cell-level and slice-level datasets are collected. Second, data is pre-processed to scale input features and create training and test datasets. Then, performance models are created by applying SL to the corresponding training dataset. Finally, model performance is assessed on test datasets. These steps are detailed next.

### A. Data collection

Tables II and III break down the set of features considered as candidate inputs for estimating DL cell throughput and DL slice throughput, respectively. Features in both tables are similar, but aggregated on a different basis (i.e., a datapoint per cell in Table II, a datapoint per cell and slice in Table III).

TABLE II: Candidate features for cell throughput estimation.

| | Feature name | Description |
|---|---|---|
| Quality | $avg\_CQI$ | Average of DL CQI distribution in the cell |
| | $median\_CQI$ | Median of DL CQI distribution in the cell |
| | $p5\_CQI$ | $5th$-tile of DL CQI distribution in the cell |
| Traffic | $avg\_actUE$ | Avg. no. of active UEs per Transmission Time Interval (TTI) in the DL of the cell |
| | $PRButil\_rat$ | PRB utilization ratio in PDSCH in the cell |
| | $s\_UE\_rat$ $\forall s \in \{VoIP, video, ftp, web\}$ | Ratio of UEs in the cell demanding each service $s$ offered in the network |
| CMs | $cell\_BW$ [MHz] | Cell bandwidth |
| | $nPRB\_i$ $\forall i = 1, 2...N_{slices}$ | No. of PRBs allocated per slice in the cell in the DL (only for NS scenarios) |

TABLE III: Candidate features for slice throughput estimation.

| | Feature name | Description |
|---|---|---|
| Quality | $avg\_CQI\_slice$ | Avg. of DL CQI distribution in the cell for UEs in the slice |
| | $median\_CQI\_slice$ | Median of DL CQI distribution in the cell for UEs in the slice |
| | $p5\_CQI\_slice$ | $5th$-tile of DL CQI distribution in the cell for UEs in the slice |
| Traffic | $avg\_actUE\_slice$ | Avg. no. of active UEs per TTI in the DL of the the cell that are served by the slice |
| | $PRButil\_rat\_slice$ | PRB utilization ratio in PDSCH considering only those PRBs allocated to the slice |
| | $s\_UE\_rat\_slice$ $\forall s \in \{VoIP, video, ftp, web\}$ | Ratio of UEs in the cell served by the slice demanding each service $s$ offered in the network |
| CMs | $cell\_BW$ [MHz] | Cell bandwidth |
| | $nPRB\_slice$ | No. of PRBs allocated to the slice in the cell |

The considered features include a) general configuration parameters (i.e., cell bandwidth), b) for NS scenarios, NS-related configuration parameters (i.e., number of PRBs allocated per slice), c) performance metrics related to spectral efficiency (i.e., CQI indicators) and d) traffic indicators (i.e., traffic mix and number of active UEs). All these features can be computed from cell-level or cell-slice-level CMs/PMs except to traffic mix features, which must be computed by aggregating information in UE-level connection traces on a cell or cell-slice basis. In live networks, PMs, CMs and traces are stored in the OSS after every Reporting Output Period (ROP, typically 15 min [29] [31]).

### B. Data pre-processing

To ensure high accuracy and faster convergence of SL algorithms, data must be scaled, so that all input features have similar ranges. For this purpose, a Z-score standardization method is used [45]. The scaled value of a certain feature, denoted as $f_{\text{scaled}}$, is computed as

$$f_{\text{scaled}} = \frac{f - \mu}{\sigma}, \tag{5}$$

where $f$ is the original feature value, and $\mu$ and $\sigma$ are the average and standard deviation, respectively, for feature $f$ in all datapoints in the dataset. Next, data is split into training and test subsets by creating a random partition, so that the training set comprises 70 % of samples and the test set includes the remaining 30 %.

### C. Model creation

The following paragraphs detail the selected regression algorithms, the dimensionality reduction scheme and the hyperparameter optimization policy.

*1) Regression algorithms:* Seven well-known SL algorithms are compared, namely Support Vector Regression (SVR), $k$-Nearest Neighbors (KNN), three ensemble methods based on Decision Trees (DTs) and 2 Artificial Neural Networks (ANNs) based on Multi-Layer Perceptrons (MLPs). Some of these algorithms have not been used to estimate throughput even in legacy cellular networks.

SVR captures non-linear relationships between features by mapping inputs into a higher dimensional feature space with a *kernel* function. Unlike classical MLR, SVR neglects deviations below an error tolerance, $\epsilon$, when finding the best regression hyperplane. To avoid overfitting, the absolute value of regression coefficients is restricted by the regularization parameter, $C$ [46].

KNN is a distance-based method which estimates the response variable of an observation by computing the averaging (often weighted) of the $k$ nearest neighbors datapoints in the training dataset according to some previously defined distance metric (e.g., Euclidean distance). It is a fast and easy-to-adjust algorithm [46].

Ensemble methods combine the output of several weak learners to perform a more robust regression. Such learners are commonly DTs, representing a flow-chart model that infer simple decision rules from the training dataset. To avoid overfitting, DTs are pruned. Three ensemble methods based on DTs are considered in this work: Random Forest (RF), Adaptive Boosting (AdaBoost) and eXtreme-Gradient Boosting (XGBoost). In RF, independent DTs are trained with different subsets of input features (a.k.a., bagging) and datapoints (a.k.a., bootstrapping), and then the output of all DTs are averaged. In contrast, in AdaBoost and XGBoost, DTs are created sequentially, so that DT$_i$ tries to improve model performance obtained with the combined method including DT$_1$ to DT$_{i-1}$ (a.k.a., boosting). For this purpose, AdaBoost increases the weight of samples with high error, and decreases the weight of samples with low error. Alternatively, in XGBoost, the gradient descent optimization algorithm is used to minimize a differentiable loss function. In both AdaBosst and XGBoost, the final output is the weighted sum of all DTs [47].

Finally, ANNs are statistical learning structures where neurons grouped in layers perform non-linear computations through activation functions [48]. Feed-forward ANNs (i.e., without memory) based on MLPs are commonly used for regression problems [49]. In this work, two MLPs are considering, differing in the number of hidden layers. The first one, denoted as Shallow MLP (SMLP), has a single hidden layer, whereas the second one, denoted as Deep MLP (DMLP), has 2 hidden layers.

Each SL algorithm is used for two purposes: a) to derive a single model to estimate DL cell throughput in any cell of the scenario, and b) to derive a single model to estimate DL slice throughput per cell in any slice of the scenario. Two options are considered for each SL algorithm and output feature (i.e., DL cell or slice throughput): a) a full model with all the candidate input features introduced in section V-A, and b) a model with a representative subset of input features selected with a wrapper method.

*2) Dimensionality reduction:* SL algorithms may underperform when input features are strongly correlated or are not relevant for the output variable. Likewise, when it comes to cellular networks, it is convenient to gather only useful information in the OSS to avoid a) congestion problems in the backhaul due to the flow of information sent from base stations, b) unnecessary investment in large databases, and c) large data pre-processing and model training times, which can be critical for real-time applications [14]. Additionally, in NS scenarios, gathering some information (e.g., connection traces) may require an agreement between tenants and infrastructure owner. As a consequence, dimensionality reduction is critical in this work. This task is performed through feature selection, which eliminates the need for gathering irrelevant indicators in the OSS (note that this issue is not solved by feature extraction) [50]. Specifically, a wrapper method denoted as Recursive Feature Elimination (RFE) is used [51]. RFE starts with a model including all the candidate input attributes, and sequentially removes the least relevant feature according to a predefined loss function. This process is repeated until an empty model is created.

In this work, RFE is performed independently per scenario, output variable and SL algorithm. The loss function is the Mean Absolute Error, $MAE$, defined as

$$MAE = \frac{1}{N_s}\sum_{i=1}^{N_s} \left|\widehat{y}(i) - y(i)\right| , \qquad (6)$$

where $N_s$ is the number of samples in the dataset, and $y(s)$ and $\widehat{y}(i)$ are the measured and estimated values of the output feature in sample $i$, respectively.

*3) Hyperparameter optimization:* An adequate configuration os hpermaramaters is key to make the most of SL models. However, a fine-grained parameter tuning increase training time exponentially. To save time, in this work, those hyperparameters showing a stable optimum value in some preliminary tests have been fixed. In contrast, those changing their optimum values with different scenarios, output variable and/or during the RFE process are set through a grid search in the parameter space [52]. Specifically, for each model, the best hyperparameter tuple is that minimizing the $MAE$ in the training dataset. Table IV breaks down the main hyperparameters per SL algorithm and the fixed values or parameter space considered.

TABLE IV: Hyperparameter tuning.

| | Hyperparameter name | Parameter space |
|---|---|---|
| SVR | Sensitivity, $\epsilon$ | [0.1,0.4] |
| | Regularization, $C$ | [10,200] |
| | Kernel function | {linear, radial basis, polynomial} |
| KNN | No. neighbors | [5,20] |
| | Distance metric | Euclidean (fixed) |
| XGBoost | No. trees | [50,200] |
| | Maximum depth | [5,10] |
| | No. features per tree | $N_f$ (fixed) |
| | Loss function | Squared error (fixed) |
| | Learning rate | [0.01,0.1] |
| | $\alpha, \lambda, \gamma$ | [0.01, 100] |
| AdaBoost | No. trees | [10,50] |
| | Maximum depth | [5,10] |
| | Loss function | {linear, square, exponential} |
| | Learning rate | [0.1 0.7] |
| RF | No. trees | [30,100] |
| | Maximum depth | [10,50] |
| | Bootstrapping | Enabled (fixed) |
| | No. of features per tree | $\{\sqrt{N_{feat}}, N_{feat}\}$ |
| SMLP and DMLP | No. layers (SMLP) | 3 (fixed) |
| | No. layers (DMLP) | 4 (fixed) |
| | No. neurons per hidden layers | [5,10] |
| | Activation function in hidden layers | {Rectified Linear Unit (ReLU), hyperbolic tangent, linear, sigmoid} |
| | Activation function in ouput layer | ReLU (fixed) |
| | Optimization algorithm | Adaptative moment (fixed) |
| | Loss function | $MAE$ (fixed) |
| | No. iterations | 1000 (fixed) |
| | Batch size | 64 (fixed) |
| | Train / validation split | 70 % / 30 % (fixed) |
| | Early stopping condition | Accuracy in the validation dataset does not improve in 3 epochs |

### *D. Performance evaluation*

Three Figures of Merit (FoMs) are used to assess model performance. Two FoMs aim to evaluate model accuracy. The first is the median value of Absolute Percentage Error, $mAPE$, computed as

$$mAPE\ [\%] = \text{median}\left(100 \cdot \left|\frac{\widehat{y}(i) - y(i)}{y(i)}\right|\right)\ \forall i \in [1, N_s] . \qquad (7)$$

The median (and not the mean) operation has been performed to avoid that insignificant errors in absolute terms in datapoints with very low throughput lead to very high and misleading percentage errors. The second FoM is the Mean Absolute Error Normalized to the maximum theoretical throughput in the cell/slice, $MANE$, defined as

$$MANE\ [\%] = \frac{1}{N_s}\sum_{i=1}^{N_s}\left(100 \cdot \left|\frac{\widehat{y}(i) - y(i)}{TH_{max}(k_i)}\right|\right) , \qquad (8)$$

where $TH_{max}(k_i)$ is the maximum achievable throughput in entity $k$ (i.e., cell/slice) to which datapoint $i$ belongs, $k_i$. In this work, $TH_{max}(k_i)$ is computed assuming that all PRBs in a cell or slice are allocated to UEs with the maximum CQI (i.e., 15). With the MCSs considered in the simulator, the peak throughput is 1 Mbps per PRB [42]. Under this assumption, $TH_{max}(k_i)$ can be computed from $cell\_BW$ for cells, and from $n\_PRB\_s$ for slices. Note that datapoints in the considered datasets have different bandwidth values, given by $cell\_BW$ and $n\_PRB\_slice$ values in cell-level and slice-level datasets, respectively. These features limit the capacity of the entity (i.e., cell or cell-slice), and, thus, figures of merit relying on absolute error (e.g., mean absolute error) would be dominated by datapoints from entities with the highest capacity, which is undesirable. The normalization performed in $MANE$ circumvents this issue. As shown in (8), $MANE$ is expressed as a percentage for an easier interpretation.

The third FoM is the number of input features per model, as a proxy of required storage capacity in the OSS and load in the backhaul due to data exchange.

## VI. Performance assessment

This section details the assessment of SL algorithms to estimate DL cell/slice throughput. For clarity, datasets are first introduced. Then, the assessment methodology is described. Next, results are presented. Finally, computational complexity is discussed.

### A. Dataset creation

Five datasets are generated with the simulator described in section IV considering noNS, NS_SS and NS_MS scenarios. For this purpose, 8 simulations with different traffic loads have been performed for each scenario and for two different system bandwidths (5 and 10 MHz), for a total of 48 simulations (16 simulations per scenario). Relative UE spatial distribution and traffic mix per cell remain constant across simulations, whereas the UE generation rate per cell is altered. Specifically, in simulation $i$, $\lambda_i(c)' = k_i \lambda(c)\ \forall c$, with $k_i > 0$.

A single simulation reflects 15 minutes of network activity (i.e., typical ROP). To avoid the transient effects of a cold start, a longer period is simulated and statistic collection starts once the adaptive capacity broker has reached steady state. The uneven cell service area, traffic density and service mix across the scenario result in each cell having different characteristics, providing realistic datasets that support the significance of results. As an example of the diverse network conditions considered, Fig. 2 shows the Cumulative Distribution Function (CDF) of $avg\_activeUE$ obtained per simulation in NS_SS scenario with a 10 MHz bandwidth. Each line comprises 108 points reflecting the average number of UEs per cell during 15 minutes of network time. It is clearly observed that cells are unevenly loaded.

Connection traces, CMs and PMs are collected during simulations. CMs and PMs are collected on a cell basis (i.e. a value per cell) in all scenarios, and on a slice basis (i.e., a value per cell and slice) in NS scenarios. All the information is grouped in five datasets depending on the scenario and if information is saved on a cell or slice basis. These datasets are denoted as NS_SS_cell, NS_MS_cell, noNS_cell, NS_SS_slice and NS_MS_slice, where the prefix denotes the scenario (i.e., noNS, NS_SS or NS_MS) and the suffix indicates if the dataset contains cell-level or slice-level information.

Each cell-level dataset contains 1,728 datapoints (i.e., 2 BWs · 8 simulations · 108 cells) with the following information:

1) Simulation index.
2) Cell identifier ($cell\_ID$).
3) The set of 14 features shown in Table II, as candidate predictors for DL cell throughput.
4) The average DL cell throughput, $TH_{cell}$, defined as the total data volume transmitted per second at the Packet Data Convergence Protocol (PDCP) layer in active periods in the DL of a cell, expressed in kbps.

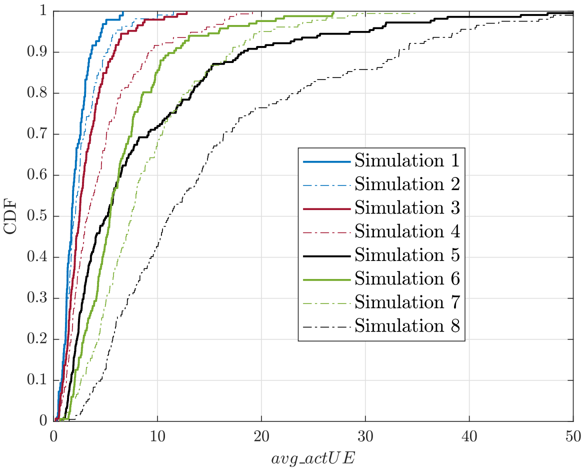

Fig. 2: CDF of number of active UEs across simulations – NS_SS scenario.

Likewise, each slice-level dataset is made of 6,912 datapoints (i.e., 2 BWs · 8 simulations · 108 cells · 4 slices) including the following information:

1) Simulation index.
2) Cell identifier ($cell\_ID$).
3) Slice identifier ($slice\_ID$).
4) The set of 10 features presented in Table III, as candidate predictors for DL slice throughput.
5) The average DL slice throughput in a certain cell, $TH_{slice}$, defined as the total data volume transmitted per second at the PDCP layer in active periods in the DL of a cell in PRBs assigned to a specific slice, expressed in kbps.

The main statistics per dataset are presented in the appendixof this manuscript (i.e., Tables XII and XIII).

### B. Assessment methodology

Three experiments are carried out.

*1) Experiment 1 – preliminary correlation analysis*: the aim of this experiment is to justify the need for deriving specific models to estimate DL cell throughput in NS scenarios. For this purpose, the average Spearman's rank correlation value, $\rho$, among several candidate input features (specifically $cell\_BW$, $PRButil\_rat$, $avg\_CQI$ and $nPRB\_slice$) and $TH_{cell}$ in noNS_cell, NS_SS_cell and NS_MS_cell datasets is compared (obviously, $nPRB\_slice$ feature is considered only in NS scenarios). $\rho$ assesses the strength and direction of monotonic association (whether linear or not) between 2 variables [53].

*2) Experiment 2 – estimation of $TH_{cell}$ in NS scenarios*: the goal of this experiment is to assess the performance of SL algorithms to estimate $TH_{cell}$ from input features in Table II in the two considered NS scenarios (i.e., single-service and multi-service slices). For this purpose, NS_SS_cell and NS_MS_cell datasets are used. As explained in section V-C, two models are created for each SL algorithm (i.e., RF, KNN, SVR...)

corresponding to each NS scenario (i.e., NS_SS and NS_MS): a) a full model with all input features (hereafter, FULL model) and b) a simplified model with input features selected by a RFE method (hereafter, RFE model).

*3) Experiment 3 (estimation of $TH_{slice}$ in NS scenarios*: this experiment is similar to experiment 2, but the aim here is to estimate slice throughput from features in slice-level datasets (i.e., NS_SS_slice and NS_MS_slice). In each scenario, the slice-level model is trained with datapoints from all slices in all cells (i.e., single output model). This model is then exploited on a cell and slice basis.

SL algorithms are implemented with *scikit-learn*, *XGBoost* and *Keras*, three machine learning libraries for Python extensively used in many fields. For further information, the reader is referred to [54] [55] [56]. A total of 56 models (=7 algorithms · 2 outputs · 2 scenarios · 2 feature sets) are tested (considering the RFE process as a single model). To prevent overfitting, a 5-fold cross validation is performed over the training dataset when tuning hyperparameters, and over the whole dataset at each step of the RFE process [46]. In the RFE process, the best number of features in the simplified model, $N_f^{opt}$, for a given SL algorithm is the minimum number of predictors achieving similar performance to the most accurate model (i.e., difference of both $mAPE$ and $MANE$ lower than 2 % in absolute terms).

A model is considered acceptable to estimate cell/slice throughput if $mAPE$<10% and $MANE$<10%. These values provide a trade-off between model complexity and accuracy [14][19]. A higher accuracy implies using complex models requiring large datasets and higher training times. On the contrary, a too relaxed threshold could lead to SLA violations if performance is overestimated, or to unnecessary actions (e.g., spectrum re-distribution among slices, bandwidth extension ...) if performance us underestimated. In live networks, accuracy threshold is up to the MNO. No matter the set value, the worst case is not detecting problems due to overestimating cell / slice performance, since user experience may be degraded. To avoid such an issue, it is key to set parameters of re-dimensioning NFs (e.g., thresholds to trigger minor, major or critical alarms) taking into consideration the expected model error to ensure that all potential problems are detected.

For robustness, the best model for each output feature is selected as follows. First, models providing $MANE$ similar to the most accurate model (i.e., difference lower than 1 % in absolute terms) are selected as candidates. Then, models with a difference in $mAPE$ higher than 1 % compared to the best $mAPE$ among candidates are discarded. Finally, the model with the lowest number of features among the remaining candidates is selected as best model. If several models satisfy this condition, the best model is that providing the best results for the worst samples (i.e., lowest 90-th error percentile). This process guarantees that the choice of the best model does not rely exclusively on $MANE$ or $mAPE$, that may be dominated by datapoints with large/small output values, respectively.

TABLE V: Correlation between input features and DL cell throughput in different scenarios.

| KPI | noNS | NS_SS | NS_MS |
|---|---|---|---|
| $cell\_BW$ | 0.59 | 0.69 | 0.70 |
| $PRButil\_rat$ | 0.04 | 0.32 | 0.36 |
| $avg\_CQI$ | 0.66 | 0.14 | 0.12 |
| $nPRB\_1$ | – | 0.66 | 0.67 |
| $nPRB\_2$ | – | 0.48 | 0.69 |
| $nPRB\_3$ | – | 0.55 | 0.68 |
| $nPRB\_4$ | – | 0.67 | 0.70 |

### *C. Results – Experiment 1 (correlation analysis)*

Table V shows the average value of Spearman's correlation coefficient, computed between $cell\_BW$, $PRButil\_rat$ and $avg\_CQI$ and $TH_{cell}$ in noNS_cell, NS_SS_cell and NS_cell datasets (e.g., average correlation between $cell\_BW$ and $TH_{cell}$ in noNS_cell dataset is 0.59). Test significance has been checked for every feature and scenario. Those features provide information about cell resources, radio resource utilization and spectral efficiency, respectively. For NS scenarios, the correlation between the number of PRBs allocated per slice, $nPRB\_s_k\ \forall\ k \in [1,4]$, and $TH_{cell}$ is also included as a metric of spectrum split. It is observed that $cell\_BW$ is significantly correlated with cell throughout in all scenarios (i.e., $\rho$≥0.59). In contrast, the correlation of $PRButil\_rat$ is much higher in NS scenarios than in non-NS scenario (i.e., $\rho$=0.32 and 0.36 in NS_SS and NS_MS scenarios, respectively, against 0.04 in noNS scenario). Likewise, $avg\_CQI$ is correlated with cell throughput in noNS scenario (i.e., $\rho$=0.66), but not in NS_SS and NS_MS scenarios ($\rho$=0.12 and 0.12, respectively). These differences point out that enabling the NS feature changes the relationships among network indicators, as anticipated in section III, revealing the need to create new performance models for NS scenarios.

It is also remarkable that similar correlation values are obtained in both NS scenarios for all features but $nPRB\_k$. This feature presents similar correlation values in all slices in NS_MS scenario (slices offering a service mix), but not in NS_SS scenario (slices offering a single service). To capture these peculiarities, specific performance models are derived per scenario in experiments 2 and 3.

### *D. Results – Experiment 2 (cell throughput)*

Tables VI and VII break down results obtained when estimating $TH_{cell}$ in NS_SS and NS_MS scenarios, respectively. Performance from FULL models is first analyzed. For a given scenario, KNN is the worst algorithm, with unacceptable $mAPE$ values. XGBoost is the best ensemble method, whereas both ANNs (SMLP and DMLP) perform similarly. In NS_SS scenario, XGBoost and ANNs are the best FULL models, with $mAPE < 7\%$ and $MANE < 2\%$. In contrast, in NS_MS scenario, ANNs outperform the rest of algorithms, with $mAPE < 5\%$ and $MANE < 1.5\%$. Comparing scenarios, similar performance (i.e., differences smaller than 2 % in absolute terms) can be obtained to estimate $TH_{cell}$ in NS_SS and NS_MS scenarios with ANNs.

TABLE VI: Model performance for estimating cell throughput in single-service NS scenario (NS_SS).

| Model | FULL | | RFE | | |
|---|---|---|---|---|---|
| FoM | $mAPE$ | $MANE$ | $N_f^{opt}$ | $mAPE$ | $MANE$ |
| SVR | 8.97 | 2.33 | 3 | 8.02 | 1.96 |
| KNN | 15.06 | 3.88 | 3 | 7.74 | 2.01 |
| XGBoost | 6.30 | 1.86 | 3 | 7.83 | 2.32 |
| AdaBoost | 8.91 | 2.14 | 4 | 8.34 | 2.15 |
| RF | 8.75 | 2.23 | 4 | 7.61 | 2.01 |
| SMLP | 6.47 | 1.71 | 3 | 7.96 | 1.89 |
| DMLP | 6.62 | 1.72 | 3 | 7.63 | 1.96 |

TABLE VII: Model performance for estimating cell throughput in multi-service NS scenario (NS_MS).

| Model | FULL | | RFE | | |
|---|---|---|---|---|---|
| FoM | $mAPE$ | $MANE$ | $N_f^{opt}$ | $mAPE$ | $MANE$ |
| SVR | 6.94 | 1.97 | 8 | 6.41 | 1.85 |
| KNN | 14.19 | 4.23 | 4 | 6.83 | 2.06 |
| XGBoost | 6.65 | 1.92 | 4 | 6.41 | 1.80 |
| AdaBoost | 7.03 | 2.07 | 9 | 7.03 | 1,96 |
| RF | 7.04 | 2.02 | 5 | 6.30 | 1.84 |
| SMLP | 4.93 | 1.40 | 4 | 5.00 | 1.38 |
| DMLP | 4.88 | 1.35 | 5 | 5.41 | 1.57 |

Fig. 3 and 4 show $MANE$ obtained across the feature selection process carried out to obtain RFE models in NS_SS and NS_MS scenarios, respectively. For a better visualization, only SVR, KNN and the best ANN and ensemble method are included. $mAPE$ evolution, not shown here for brevity, is very similar. As expected, for most algorithms, the larger the number of input features, the better accuracy, until adding a new feature does not improve model performance significantly. It is remarkable that, in both scenarios, KNN performance degrades when the model is trained with more than 3 features. This behavior reveals that KNN is suffering the so called *curse of dimensionality*, since it requires all neighbor datapoints to be close in all dimensions of the data space, which becomes more difficult as the input feature space grows [57]. Such a phenomenon is also observed in SVR for $N_f \in \{3, \ldots, 7\}$ in Fig. 4.

The last three columns in Table VI summarize FoMs with the best RFE model for all tested algorithms in NS_SS scenario. $N_f^{opt}$ is selected with the convergence criteria described above, resulting $N_f^{opt} = 3$ for all algorithms but AdaBoost and RF, with $N_f^{opt} = 4$. RFE models requiring 3 input features (i.e., all but AdaBoost and RF) are considered as potential candidates to estimate $TH_{cell}$. Since accuracy of both ANNs is similar and SMLP is faster to train and less prone to overfitting than DMLP, DMLP is discarded. For a deeper analysis, Fig. 5 shows the Cumulative Function Distribution (CDF) of $ANE$ obtained with the remaining candidate models. All algorithms perform similar in the lower part of the CDF, whereas SMLP performs best in the upper part (lowest errors). Thus, RFE–SMLP is considered the best model. The selected input features (ranked by relevance) are $PRButil\_rat$, $avg\_CQI$ and $cell\_BW$.

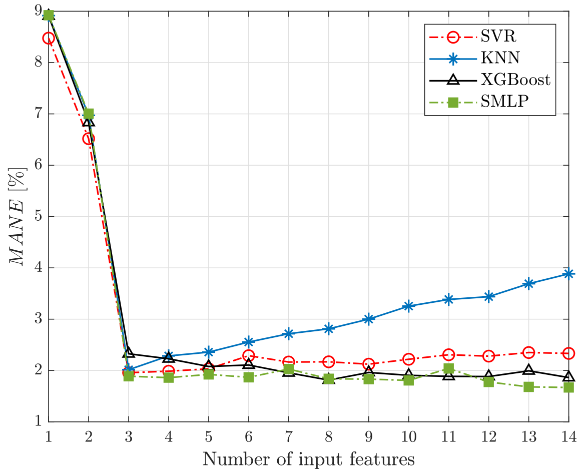

Fig. 3: $MANE$ evolution across RFE process when estimating cell throughput in single service NS scenario (NS_SS).

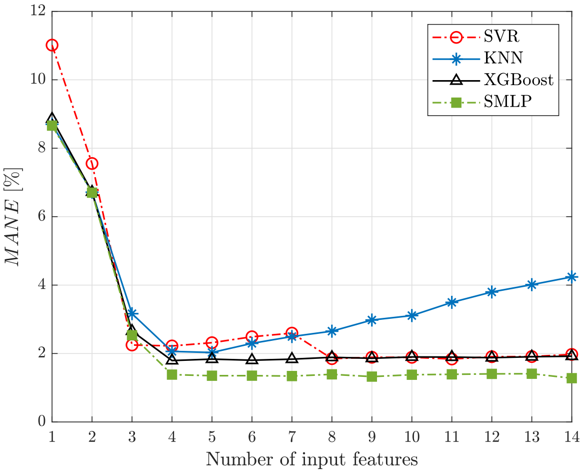

Fig. 4: $MANE$ evolution across RFE process when estimating cell throughput in multi-service NS scenario (NS_MS).

Similarly, the last three columns in Table VII summarize FoMs obtained with the best RFE models in NS_MS scenario. Unlike in NS_SS scenario, the algorithms now have completely different $N_f^{opt}$, ranging from 4 to 9. The best model is RFE–SMLP, with the highest accuracy (i.e., $mAPE = 5\%$ and $MANE = 1.38\%$) and the lowest number of input features (i.e., $N_f^{opt} = 4$). Not shown in the table is the fact that some of the relevant input features are also different, with $PRButil\_rat$, $median\_CQI$, $cell\_BW$ and $p5\_CQI$ (listed by relevance). The significant decrease in $MANE$ for SMLP from $N_f$=3 to $N_f$=4 observed in Fig 4 confirms that, in NS_MS scenario, the inclusion of spectral efficiency of cell-edge UEs through $p5\_CQI$ improves SMLP performance. Such an effect may be due to joint packet scheduling for UEs demanding different services in each slice, which favors cell-edge UEs from services with strict delay requirements (e.g., VoIP) at the expense of UEs with better channel conditions from services with loose delay constraints, decreasing cell throughput even with a high $PRButil\_rat$.

From the above results, it can be concluded that SMLP

This article has been accepted for publication in IEEE Transactions on Network and Service Management. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/TNSM.2022.3208336

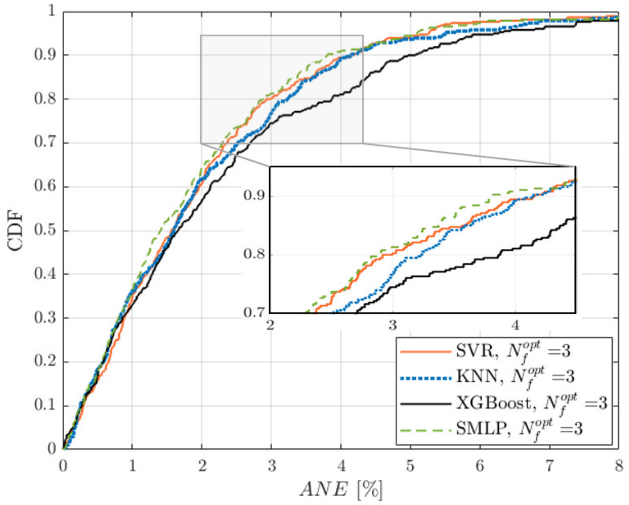

Fig. 5: Distribution of error metrics for best models when estimating cell throughput in single-service NS scenario (NS_SS).

TABLE VIII: Model performance for estimating slice throughput in single-service NS scenario (NS_SS).

| Model | FULL | | RFE | | |
|---|---|---|---|---|---|
| FoM | $mAPE$ | $MANE$ | $N_f^{opt}$ | $mAPE$ | $MANE$ |
| SVR | 14.40 | 8.30 | 10 | 12.64 | 7.81 |
| KNN | 14.56 | 9.84 | 3 | 8.92 | 3.96 |
| XGBoost | 7.37 | 2.87 | 4 | 8.16 | 3.05 |
| AdaBoost | 6.44 | 2.86 | 7 | 8.15 | 3.21 |
| RF | 5.53 | 2.57 | 5 | 6.26 | 2.75 |
| SMLP | 7.70 | 3.27 | 4 | 7.80 | 2.97 |
| DMLP | 6.78 | 2.83 | 5 | 9.63 | 4.46 |

TABLE IX: Model performance for estimating slice throughput in multi-service NS scenario (NS_MS).

| Model | FULL | | RFE | | |
|---|---|---|---|---|---|
| FoM | $mAPE$ | $MANE$ | $N_f^{opt}$ | $mAPE$ | $MANE$ |
| SVR | 12.47 | 7.32 | 4 | 12.51 | 7.22 |
| KNN | 16.30 | 9.73 | 3 | 12.08 | 6.99 |
| XGBoost | 9.64 | 5.58 | 6 | 9.46 | 5.53 |
| AdaBoost | 10.24 | 5.77 | 8 | 9.86 | 5.53 |
| RF | 9.34 | 5.48 | 6 | 9.16 | 5.70 |
| SMLP | 8.82 | 5.10 | 5 | 8.78 | 5.25 |
| DMLP | 8.87 | 5.17 | 5 | 9.23 | 5.45 |

is an adequate SL algorithm to estimate cell throughput in both NS scenarios, providing acceptable accuracy with models requiring few input features. With an adequate feature selection, similar accuracy can be obtained for estimating $TH_{cell}$ in NS scenarios with single-service or multi-service slices, with an error lower than 2% of the achievable cell throughput. It should be pointed out that, in both scenarios, the input features to RFE–SMLP model are similar to those in classical models, shown in (1). Those features can be computed from PMs/CMs stored in the OSS for most network operators. Thus, it is not necessary to store additional NS-specific information or to collect connection traces to estimate cell throughput in NS scenarios. A deeper analysis of data shows that, in both scenarios, the worst datapoints (i.e., those with highest $ANE$) correspond to underutilized cells with high spectral efficiency (i.e., $PRButil\_rat$ < 40%, $avg\_CQI \geq 10$), where $TH_{cell}$ tends to be underestimated.

### *E. Results – Experiment 3 (slice throughput)*

Tables VIII and IX summarize results when estimating $TH_{slice}$ in NS_SS and NS_MS scenarios, respectively. Likewise, Fig. 6 and 7 present the evolution of $MANE$ across RFE for all algorithms in these scenarios (DMLP lines have been omitted for a better visualization, since they overlap with SMLP).

For FULL models in both scenarios, only algorithms based on DTs and ANNs fulfill the accuracy threshold of 10 % for both $mAPE$ and $MANE$. RF provides the best FULL model in NS_SS scenario ($mAPE$=5.53 % and $MANE$=2.57 %), whereas both ANNs get the best results in NS_MS scenario ($mAPE \approx 8.85$ % and $MANE \approx 5.15$ %). When comparing FULL and RFE models for each scenario, it is observed that, again, for a given algorithm, similar performance can be obtained with simpler models with less input variables.

Regarding RFE, Fig. 6 and 7 reveal that KNN is again suffering the curse of dimensionality, since accuracy diminishes when increasing the number of features above $N_f = 3$. It is also remarkable that the evolution of SVR performance in NS_MS scenario, with $N_f^{opt} = 4$, differs significantly from NS_SS scenario, where only one feature can be extracted for an acceptable model performance (i.e., $MANE$ increases significantly below $N_f$=10). Nonetheless, unlike when predicting $TH_{cell}$, SVR is not competitive with other SL algorithms.

When considering a trade-off between accuracy and input size, RFE–RF is the best model in NS_SS scenario ($mAPE$=6.26% and $MANE$=2.75% with $N_f^{opt}$=5), followed by RFE–SMLP. In NS_MS scenario, RFE models built with ANNs show the best accuracy (i.e., $mAPE \approx 9\%$ and $MANE \approx 5.35\%$) and required information ($N_f^{opt} = 5$). Fig. 8 represents $ANE$ CDFs obtained with these models. RFE–XGBoost model (i.e., the next model with better $MANE$) is also included. A significant improvement of ANNs over XGBoost is observed for the largest error percentiles. Among ANNs, RFE–SMLP is the best option (lines are shifted to the left compared to RFE—DMLP). The input features in the best models (i.e., RFE–RF for NS_SS scenario and RFE–SMLP for NS-MS scenario) are $cell\_BW$, $nPRB\_slice$, $PRButil\_rat\_slice$, $median\_CQI\_slice$ and $voip\_UE\_rat\_slice$ in both scenarios.

Table X shows the results for the best models broken down per slice. Recall that, in NS_SS scenario, slices 1 to 4 serve UEs demanding VoIP, video, file download and web browsing, respectively. In contrast, in NS_MS scenario, slices serve a service mix changing with cell and tenant. It can be noticed that differences among slices are larger in NS_SS than in NS_MS scenario. For a more detailed analysis, Fig. 9 and 10 show the probability density function of $PRButil\_rat\_slice$ and $slice\_ID$ for 5 % of samples with the largest error (i.e., highest $MANE$) for the best models in NS_SS and NS_MS scenarios. In NS_SS scenario, $slice\_ID$ distribution

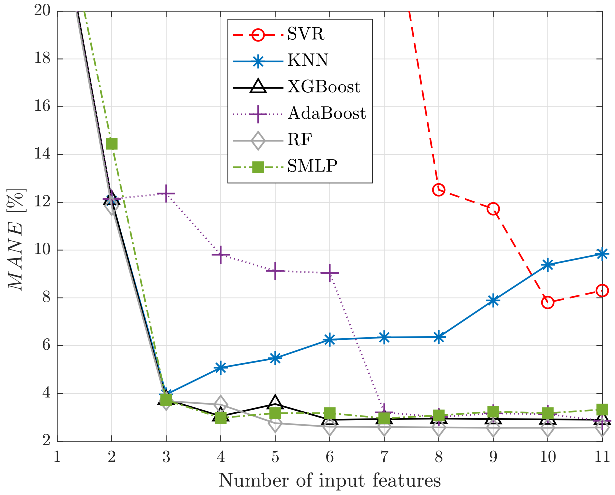

Fig. 6: $MANE$ evolution across RFE process when estimating slice throughput in single-service scenario (NS_SS).

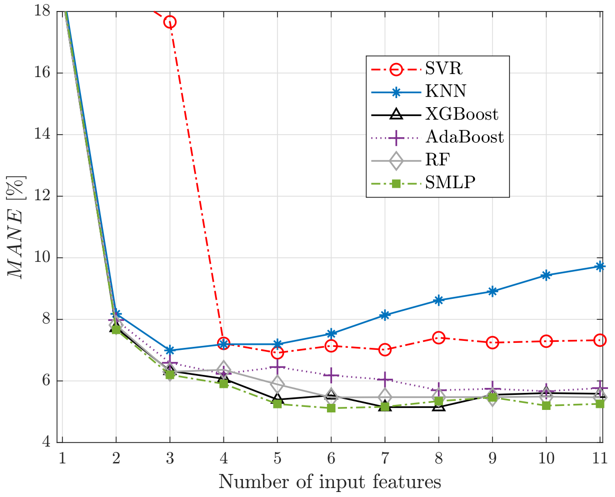

Fig. 7: $MANE$ evolution across RFE process when estimating slice throughput in multi-service NS scenario (NS_MS).

reveals that more than 80 % of worst samples are from slice 3 to 4. Thus, RFE–RF provides the lowest accuracy when estimating throughput from slices serving web and file download users. This behavior is observed in most of the tested SL algorithms. Note that the data rate of users of these best effort services adapts to instantaneous slice capacity (i.e., PRBs allocated) and traffic (i.e., UEs to schedule) in the cell and is thus prone to fluctuate, being more difficult to estimate. This problem may be solved by creating per-service slice-level models at the expense of having less training datapoints per model. In NS_MS scenario, worst offenders for RFE–SMLP are evenly distributed among slices. However, it is remarkable that 70 % of these datapoints belong to slices with $PRButil\_rat\_slice \leq 20\%$. Thus, it can be concluded that RFE–SMLP is less accurate when predicting the aggregate throughput of underutilized slices, no matter the slice service mix.

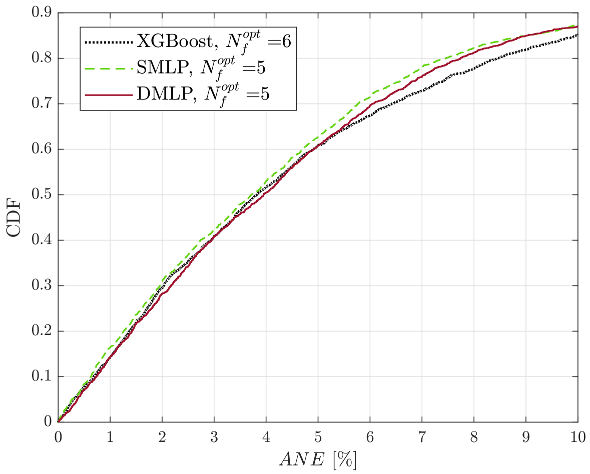

Fig. 8: Distribution of absolute normalized error for best models when estimating slice throughput in multi-service NS scenario (NS_MS).

TABLE X: Performance per slice when estimating slice throughput with the best model.

| Scenario | NS_SS | | NS_MS | |
|---|---|---|---|---|
| FoM | $mAPE$ | $MANE$ | $mAPE$ | $MANE$ |
| Slice 1 | 14.43 | 0.38 | 9.99 | 5.19 |
| Slice 2 | 4.82 | 3.09 | 9.41 | 5.34 |
| Slice 3 | 4.74 | 3.49 | 6.97 | 5.35 |
| Slice 4 | 5.51 | 3.86 | 8.92 | 5.25 |

From the above results, it can be concluded that the best SL algorithm to estimate slice throughput depends on the NS scenario. Moreover, it is worth noting that, unlike when estimating cell throughput, accuracy obtained with the best model is lower in NS_MS scenario ($MANE$=5.25%) than in NS_SS scenario ($MANE$=2.75%). This may be due to the coexistence of users with services with very different traffic patterns, which makes throughput calculations more complex. Likewise, the RFE process shows that it is convenient to use information about the service mix to estimate $TH_{slice}$. Specifically, $voip\_UE\_rat\_slice$ has been selected as key feature in this work. Note that VoIP is the service with the lowest data rate in the considered scenario. Thus, this feature provides information about the ratio of data-hungry UEs in the slice. This information may be useful to estimate throughput since bursty traffic degrades network spectral efficiency due to last transmission time interval data and outer loop link adaptation [58]. In NS_SS scenarios, slice service mix is simple (i.e., no mix), whereas in NS_MS scenarios it can be obtained by applying a traffic classification algorithm over radio connection traces, even if traffic is encrypted [59].

## *F. Computational complexity*

The implementation of SL models for estimating throughput in radio planning tools or RRM NFs entails: a) collecting and pre-processing data in the OSS, b) selecting the best model

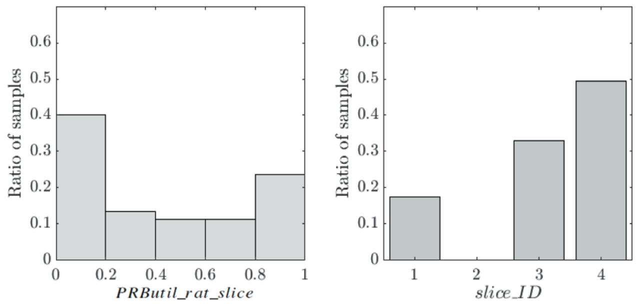

Fig. 9: Values of features in samples with the largest error when estimating slice throughput with RFE–RF (NS_SS scenario).

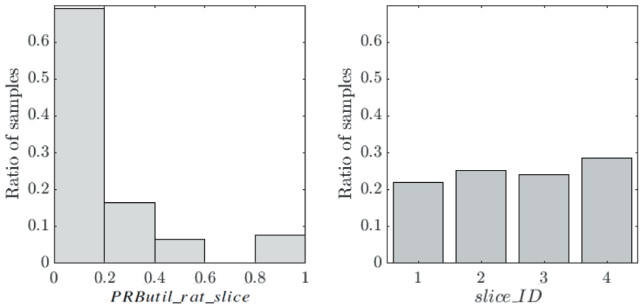

Fig. 10: Values of features in samples with the largest error when estimating slice throughput with RFE–SMLP (NS_MS scenario).

(i.e., combination of SL algorithm and set of input features) for the specific network, c) exploiting the model and d) retraining the model when necessary.

Data used to compute the considered input features (i.e., PMs/CMs/traces) is often collected by network operators for network management purposes. Processing radio connection traces can be a time-consuming task. According to the results in this work, this task has to be performed only for slice-level performance models in multi-service slice scenarios. In this case, parallelization can be used to speed up trace processing.

Finding the best model implies to carry out the RFE process for one or several candidate SL algorithms. The reader is referred to [60] and [14] for the analytical worst-case time complexity of XGBoost and the remaining tested algorithms, respectively. For instance, Table XI shows training time for the FULL models when estimating $TH_{slice}$ in NS_SS scenario (i.e., the case with the largest number of datapoints) in a personal computer with Intel Core i7-8700 processor working at 3.2 GHz with a RAM of 16 GB. Training times range from few seconds to near 9 min. Differences are due to algorithm complexity and specially to the varying number of hyperparameters tested per algorithm. Training time decreases significantly with RFE models. Nonetheless, in case of strict time constrains, hyperparameter optimization can be accelerated via parallelization or relaxed (e.g., running less folds of data). Since the latter option may degrade model performance, the former alternative is preferred when necessary. Once the best SL scheme is selected, exploiting the models is immediate. Specifically, in this work, prediction time per datapoint is approximately 0.5 ms. Such a time meets the requirements even of the most stringent slice re-dimensioning NFs, typically working on a second or millisecond timescale.

TABLE XI: Time complexity of FULL models when estimating slice throughput in single-service NS scenario.

| Algorithm | Training time [s] |
|---|---|
| SVR | 114 |
| KNN | 0.6 |
| XGBoost | 458 |
| AdaBoost | 13 |
| RF | 27 |
| SMLP | 164 |
| DMLP | 864 |

Both slice and cell level models must be executed again after any significant change affecting input variables (e.g., change in traffic demand, traffic mix, radio channel conditions or cell bandwidth). Moreover, slice-level estimates must be updated if PRB split among slices varies due to SLA violation, SLA re-definition or slice activation/de-activation. Likewise, models must be retrained if an event changing the relationship between predictors and the output variable happens in the network (e. g., an update of packet scheduling or capacity broker policy, the launch of new services or the introduction of new terminal and base station capabilities). New models must also be trained for a different network.

## VII. Conclusions

In 5G radio access networks, the introduction of the Network Slicing (NS) makes legacy network performance models vary significantly. In this new scenario, slice-level and cell-level throughput estimates are required for network management purposes, such as cell re-planning or spectrum sharing among slices. To this end, supervised learning is a promising solution to derive performance models tailored to specific network architectures and NS implementations. In this work, a comprehensive analysis has been carried out to assess the performance of 7 well-known SL algorithms for estimating cell and slice throughput in the DL from radio network performance counters and connection traces collected in the OSS. Performance assessment has been carried out over cell-level and slice-level datasets built with a system-level simulator. This tool a) is dynamic (i.e., a simulation consists of a set of correlated snapshots emulating network activity along time on a 10-ms resolution), b) includes realistic traffic models from 4 different services, c) implements real RRM algorithms from vendors and d) considers a realistic scenario comprising 108 cells with uneven cell service area, traffic density and service mix. All these characteristics allow to provide realistic datasets that support the significance of results. Two different NS scenarios are considered, with single-service slices and multi-service slices. The proposed models are conceived to be used in centralized solutions running in the OSS, where data from all cells is available.

Results show that, with the adequate feature selection process, all the tested algorithms achieve acceptable performance (i.e., $MANE$ and $mAPE$ lower than 10%) when estimating cell throughput in NS scenarios, using similar information to models in non-NS scenarios. Moreover, all algorithms perform similarly in scenarios with single-service and multi-service slices. When considering the trade-off between accuracy and storage capacity, SMLP has shown the best results in both scenarios, with $MANE$=2% and $mAPE$¡8%. The optimum models have at most 4 input features related to bandwidth, radio resource utilization and spectral efficiency. Such features can be computed from PMs/CMs collected on a cell basis in the OSS.

However, when estimating slice throughput, only ensemble methods and ANNs achieve acceptable accuracy. Moreover, model accuracy is worse in multi-service slices, where users demand services with highly differing traffic patterns. RFE–RF has shown the best accuracy in single-service NS scenario ($mAPE$ = 6.6.26%, $MANE$ = 2.75%), whereas RFE–SMLP has performed best in multi-service NS scenario ($mAPE$ = 8.78%, $MANE$ = 5.25%). In both cases, the 5 input features to these models not only include indicators computed from PMs/CMs at cell level, but also indicators computed at slice level and information about the service mix per slice derived from connection traces.

Throughput models proposed in this work can be used to detect resource overprovisioning, capacity problems or SLA violations in network (re)dimensioning. For instance, cell / slice performance in the worst conditions can be checked by setting input features to their worst value (e.g., highest allowed RB utilization). Moreover, future throughput performance can be predicted by feeding the models with forecasts of input features. It should be pointed out that models do not directly provide the specific re-dimensioning action to be taken to solve the detected problem (e.g., bandwidth extension/reduction, deployment of a new carrier…). Nonetheless, models can still be used to assess the impact of candidate re-planning actions on network performance.

As future work, we propose to reproduce the presented analysis over real data when available. Then, more complex models based on DNN may be tested, which would have overfitted with the size of the simulated datasets. Additionally, we will consider the use of transfer learning [61] to leverage pre-trained models derived for slices with different RRM algorithms (e.g., slices managed by VMNOs with different packet scheduler). Likewise, the use of multitask ANNs [62] will be explored to jointly estimate the performance of multiple slices in a cell and its neighbors. Another promising research direction is including uRLLC and mMTC services to a) assess the impact on throughput modelling and b) model other performance metrics (e.g., latency-reliability for uRLLC). Note that emulating uRLLC traffic accurately implies analyzing network activity with a 1-ms time resolution. Likewise, mMTC services are characterized by an extremely high number of devices connected simultaneously to the network. These requirements would increase simulation time by more than 10 times, making unfeasible to run the 48 simulations used in this work. Thus, this study have focused on eMBB use case.

## Acknowledgements

This work has been funded by the Spanish Ministry of Science and Innovation (RTI2018-099148-B-I00 and PID2021-122217OB-I00) and the Spanish Ministry of Education, Culture and Sports (FPU17/04286).

## Appendix

Tables XII and XIII present a statistical summary of the 3 cell-level datasets and 2 slice-level datasets used in this work, respectively. For each dataset, the following information is provided: a) number of samples and b) mean, standard deviation, maximum value and minimum value of the input and output features.

## References

[1] 3rd Generation Parthnership Project, "Feasibility Study on New Services and Markets Technology Enablers," in *TS 22.891, version 14.2.0*, 2016.

[2] T. e. a. Taleb, "White paper on 6G networking," 2020.

[3] W. Saad, M. Bennis, and M. Chen, "A vision of 6G wireless systems: Applications, trends, technologies, and open research problems," *IEEE network*, vol. 34, no. 3, pp. 134–142, 2019.

[4] N. An, Y. Kim, J. Park, D.-H. Kwon, and H. Lim, "Slice management for quality of service differentiation in wireless network slicing," *Sensors (MDPI)*, vol. 19, no. 12, p. 2745, 2019.

[5] NGMN Alliance, "Description of network slicing concept," in *NGMN 5G P1: Requirements Architecture Work Stream End-to-End Architecture*, 2016.

[6] Ericsson, "Network slicing: A go-to-market guide to capture the high revenue potential," 2021.

[7] S. E. Elayoubi, S. B. Jemaa, Z. Altman, and A. Galindo-Serrano, "5G RAN slicing for verticals: Enablers and challenges," *IEEE Communications Magazine*, vol. 57, no. 1, pp. 28–34, 2019.

[8] 3rd Generation Parthnership Project, "Management and orchestration; Concepts, use cases and requirements," in *TS 28.530, version 17.1.0*, 2021.

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

[10] J. Pérez-Romero, O. Sallent, R. Ferrús, and R. Agustí, "Profit-based radio access network slicing for multi-tenant 5G networks," in *2019 European Conference on Networks and Communications (EuCNC)*, pp. 603–608, IEEE, 2019.

[11] ITU-R Recommendation M.2083, "IMT Vision – Framework and overall objectives of the future development of IMT for 2020 and beyond," 2015.

[12] J. Moysen, L. Giupponi, and J. Mangues-Bafalluy, "A mobile network planning tool based on data analytics," *Mobile Information Systems*, vol. 2017, 2017.

[13] J. A. Fernández-Segovia, S. Luna-Ramírez, M. Toril, and J. J. Sánchez-Sánchez, "Estimating cell capacity from network measurements in a multi-service LTE system," *IEEE Communications Letters*, vol. 19, no. 3, pp. 431–434, 2015.

[14] C. Gijón, M. Toril, S. Luna-Ramírez, J. L. Bejarano-Luque, and M. L. Marí-Altozano, "Estimating Pole Capacity From Radio Network Performance Statistics by Supervised Learning," *IEEE Transactions on Network and Service Management*, vol. 17, no. 4, pp. 2090–2101, 2020.

[15] B. Ma, W. Guo, and J. Zhang, "A survey of online data-driven proactive 5G network optimisation using machine learning," *IEEE Access*, vol. 8, pp. 35606–35637, 2020.

[16] M. G. Kibria, K. Nguyen, G. P. Villardi, O. Zhao, K. Ishizu, and F. Kojima, "Big data analytics, machine learning, and artificial intelligence in next-generation wireless networks," *IEEE access*, vol. 6, pp. 32328–32338, 2018.

[17] P. A. Sánchez, S. Luna-Ramírez, M. Toril, C. Gijón, and J. L. Bejarano-Luque, "A data-driven scheduler performance model for QoE assessment in a LTE radio network planning tool," *Computer Networks*, vol. 173, p. 107186, 2020.

[18] V. Wille, M. Toril, and S. Luna-Ramirez, "Estimating pole capacity in a live HSDPA network," *IEEE Communications Letters*, vol. 17, no. 6, pp. 1260–1263, 2013.

TABLE XII: Statistics of cell-level datasets used to estimate DL cell throughput.

| Dataset name | noNS_cell | | | | NS_SS_cell | | | | NS_MS_cell | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| No. datapoints | 1,728 | | | | 1,728 | | | | 1,728 | | | |
| Statistic | Mean | Std. deviation | Min. | Max. | Mean | Std. deviation | Min. | Max. | Mean | Std. deviation | Min | Max |
| $avg\_CQI$ | 10.46 | 2.85 | 1.97 | 15.99 | 10.84 | 2.63 | 3.20 | 15.97 | 10.45 | 2.61 | 3.28 | 15.87 |
| $median\_CQI$ | 10.53 | 3.45 | 1 | 16 | 11.01 | 3.31 | 2 | 16 | 10.48 | 3.29 | 1 | 16 |
| $p5\_CQI$ | 5.26 | 3.08 | 1 | 16 | 5.64 | 2.87 | 1 | 16 | 5.22 | 2.75 | 1 | 15 |
| $avg\_actUE$ | 8.24 | 24.03 | 0.26 | 55.63 | 12.00 | 25.18 | 0.48 | 58.76 | 14.32 | 25.40 | 0.65 | 62.42 |
| $PRButil\_rat$ | 0.57 | 0.33 | 0.01 | 1 | 0.43 | 0.27 | 0.02 | 1 | 0.55 | 0.30 | 0.01 | 1 |
| $voip\_UE\_rat$ | 0.22 | 0.09 | 0.04 | 0.59 | 0.22 | 0.09 | 0.04 | 0.59 | 0.22 | 0.09 | 0.04 | 0.59 |
| $video\_UE\_rat$ | 0.25 | 0.10 | 0.05 | 0.67 | 0.25 | 0.10 | 0.05 | 0.67 | 0.25 | 0.10 | 0.05 | 0.67 |
| $ftp\_UE\_rat$ | 0.26 | 0.10 | 0.02 | 0.67 | 0.26 | 0.10 | 0.02 | 0.67 | 0.26 | 0.10 | 0.02 | 0.67 |
| $web\_UE\_rat$ | 0.27 | 0.11 | 0.03 | 0.69 | 0.27 | 0.11 | 0.03 | 0.69 | 0.27 | 0.11 | 0.03 | 0.69 |
| $cell\_BW$ [MHz] | – | – | 5 | 10 | – | – | 5 | 10 | – | – | 5 | 10 |
| $nPRB\_1$ | – | – | – | – | 3.74 | 1.12 | 3 | 8 | 9.99 | 4.31 | 3 | 24 |
| $nPRB\_2$ | – | – | – | – | 20.45 | 8.19 | 4 | 40 | 10.05 | 5.09 | 3 | 27 |
| $nPRB\_3$ | – | – | – | – | 5.78 | 4.68 | 3 | 22 | 9.76 | 5.17 | 4 | 28 |
| $nPRB\_4$ | – | – | – | – | 8.81 | 8.55 | 3 | 38 | 9.54 | 4.82 | 3 | 29 |
| $TH_{cell}$ [kbps] | 14894.13 | 8119.25 | 1352.95 | 43382.93 | 8174.02 | 5088.90 | 393.24 | 29845.14 | 9973.97 | 6314.22 | 830.29 | 35819.48 |

TABLE XIII: Statistics of slice-level datasets used to estimate DL slice throughput.

| Dataset name | NS_SS_slice | | | | NS_MS_slice | | | |
|---|---|---|---|---|---|---|---|---|
| No. datapoints | 6,912 | | | | 6,912 | | | |
| Statistic | Mean | Std. deviation | Min. | Max. | Mean | Std. deviation | Min. | Max. |
| $avg\_CQI\_slice$ | 11.06 | 3.30 | 2 | 16 | 10.79 | 3.26 | 1.22 | 16 |
| $median\_CQI\_slice$ | 11.09 | 3.80 | 1 | 16 | 10.78 | 3.76 | 1 | 16 |
| $p5\_CQI\_slice$ | 7.46 | 3.88 | 1 | 16 | 7.12 | 3.81 | 1 | 16 |
| $avg\_actUE\_slice$ | 4.56 | 9.65 | 0.13 | 20.02 | 4.74 | 7.42 | 0.26 | 20.43 |
| $PRButil\_rat\_slice$ | 0.52 | 0.34 | 0.05 | 1 | 0.54 | 0.35 | 0.06 | 1 |
| $voip\_UE\_rat\_slice$ | $s\_UE\_rat\_slice$=1 for the service $s$ offered in the slice and 0 for the rest of services | | | | 0.21 | 0.22 | 0 | 1 |
| $video\_UE\_rat\_slice$ | | | | | 0.25 | 0.23 | 0 | 1 |
| $ftp\_UE\_rat\_slice$ | | | | | 0.25 | 0.24 | 0 | 1 |
| $web\_UE\_rat\_slice$ | | | | | 0.27 | 0.24 | 0 | 1 |
| $cell\_BW$ [MHz] | – | – | 5 | 10 | – | – | 5 | 10 |
| $nPRB\_slice$ | 10.94 | 9.57 | 3 | 40 | 9.89 | 4.82 | 3 | 29 |
| $TH_{slice}$ [kbps] | 5389.22 | 5763.09 | 21.01 | 35353.03 | 4370.87 | 3047.14 | 16.35 | 21772.30 |

[19] D. Parracho, D. Duarte, I. Pinto, and P. Vieira, "An improved capacity model based on radio measurements for a 4G and beyond wireless network," in *21st International Symposium on Wireless Personal Multimedia Communications (WPMC)*, pp. 314–318, 2018.

[20] A. A. Barakabitze, A. Ahmad, R. Mijumbi, and A. Hines, "5G network slicing using SDN and NFV: A survey of taxonomy, architectures and future challenges," *Computer Networks*, vol. 167, p. 106984, 2020.

[21] M. Chahbar, G. Diaz, A. Dandoush, C. Cérin, and K. Ghoumid, "A comprehensive survey on the E2E 5G network slicing model," *IEEE Transactions on Network and Service Management*, vol. 18, no. 1, pp. 49–62, 2020.

[22] M. O. Ojijo and O. E. Falowo, "A survey on slice admission control strategies and optimization schemes in 5G network," *IEEE Access*, vol. 8, pp. 14977–14990, 2020.

[23] R. Su, D. Zhang, R. Venkatesan, Z. Gong, C. Li, F. Ding, F. Jiang, and Z. Zhu, "Resource allocation for network slicing in 5G telecommunication networks: A survey of principles and models," *IEEE Network*, vol. 33, no. 6, pp. 172–179, 2019.

[24] O. Sallent, J. Perez-Romero, R. Ferrus, and R. Agusti, "On radio access network slicing from a radio resource management perspective," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 166–174, 2017.

[25] B. Han and H. D. Schotten, "Machine learning for network slicing resource management: a comprehensive survey," *arXiv preprint arXiv:2001.07974*, 2020.

[26] O. Adamuz-Hinojosa, P. Ameigeiras, P. Munoz, and J. M. Lopez-Soler, "Analytical Model for the UE Blocking Probability in an OFDMA Cell providing GBR Slices," in *2021 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1–7, IEEE, 2021.

[27] P. Muñoz, O. Sallent, and J. Pérez-Romero, "Self-dimensioning and planning of small cell capacity in multitenant 5g networks," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 5, pp. 4552–4564, 2018.

[28] V. Buenestado, M. Toril, S. Luna-Ramírez, J. M. Ruiz-Avilés, and A. Mendo, "Self-tuning of remote electrical tilts based on call traces for coverage and capacity optimization in LTE," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 5, pp. 4315–4326, 2016.

[29] C. Gijon, M. Toril, S. Luna-Ramirez, and M. L. Mari-Altozano, "A data-driven traffic steering algorithm for optimizing user experience in multi-tier LTE networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 10, pp. 9414–9424, 2019.

[30] M. U. Khan, A. García-Armada, and J. Escudero-Garzás, "Service-Based Network Dimensioning for 5G Networks Assisted by Real Data," *IEEE Access*, vol. 8, pp. 129193–129212, 2020.

[31] A. J. García, M. Toril, P. Oliver, S. Luna-Ramírez, and M. Ortiz, "Automatic alarm prioritization by data mining for fault management in cellular networks," *Expert Systems with Applications*, vol. 158, p. 113526, 2020.

[32] J. Moysen and L. Giupponi, "From 4G to 5G: Self-organized network management meets machine learning," *Computer Communications*, vol. 129, pp. 248–268, 2018.

[33] S. Zhang and D. Zhu, "Towards artificial intelligence enabled 6G: State of the art, challenges, and opportunities," *Computer Networks*, p. 107556, 2020.

[34] I. Vilà, J. Pérez-Romero, O. Sallent, and A. Umbert, "A Novel Approach for Dynamic Capacity Sharing in Multi-tenant Scenarios," in *2020 IEEE 31st Annual International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1–6, 2020.

[35] S. Matoussi, I. Fajjari, N. Aitsaadi, and R. Langar, "Deep Learning based User Slice Allocation in 5G Radio Access Networks," in *2020 IEEE 45th Conference on Local Computer Networks (LCN)*, pp. 286–296, IEEE, 2020.

[36] M. H. Abidi, H. Alkhalefah, K. Moiduddin, M. Alazab, M. K. Mohammed, W. Ameen, and T. R. Gadekallu, "Optimal 5g network

slicing using machine learning and deep learning concepts," *Computer Standards & Interfaces*, vol. 76, p. 103518, 2021.

[37] C. Baena, S. Fortes, E. Baena, and R. Barco, "Estimation of video streaming KQIs for radio access negotiation in network slicing scenarios," *IEEE Communications Letters*, vol. 24, no. 6, pp. 1304–1307, 2020.

[38] H. Wang, Y. Wu, G. Min, and W. Miao, "A graph neural network-based digital twin for network slicing management," *IEEE Transactions on Industrial Informatics*, 2020.

[39] Ericsson, "Ericsson Mobility Report," Jun. 2020.

[40] M. L. Marí-Altozano, S. S. Mwanje, S. Luna-Ramírez, M. Toril, H. Sanneck, and C. Gijón, "A service-centric Q-learning algorithm for mobility robustness optimization in LTE," *IEEE Transactions on Network and Service Management*, 2021.

[41] 3rd Generation Parthnership Project, "New Radio (NR); User Equipment (UE) radio transmission and reception; Part 1: Range 1 Standalone," in *TS 38.101-1, version 17.2.0*, 2021.

[42] 3rd Generation Parthnership Project, "New Radio (NR); Physical Layer procedures for data," in *TS 38.214, version 16.6.0*, 2021.

[43] K. Brueninghaus, D. Astely, T. Salzer, S. Visuri, A. Alexiou, S. Karger, and G.-A. Seraji, "Link performance models for system level simulations of broadband radio access systems," in *16th International Symposium on Personal, Indoor and Mobile Radio Communications*, vol. 4, pp. 2306–2311, IEEE, 2005.

[44] J.-H. Rhee, J. M. Holtzman, and D.-K. Kim, "Scheduling of real/non-real time services: adaptive EXP/PF algorithm," in *The 57th IEEE Semiannual Vehicular Technology Conference, 2003 (VTC-2003-Spring)*, vol. 1, pp. 462–466, 2003.

[45] J. Han, J. Pei, and M. Kamber, *Data mining: concepts and techniques*. Elsevier, 2011.

[46] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, *The elements of statistical learning: data mining, inference and prediction, Second edition*. Springer series in statistics, 2001.

[47] L. Rokach, "Ensemble methods in supervised learning," in *Data mining and knowledge discovery handbook*, pp. 959–979, Springer, 2009.

[48] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *13th International Conference on Artificial Intelligence and Statistics*, pp. 249–256, 2010.

[49] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[50] S. Khalid, T. Khalil, and S. Nasreen, "A survey of feature selection and feature extraction techniques in machine learning," in *2014 science and information conference*, pp. 372–378, IEEE, 2014.

[51] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *Journal of machine learning research*, vol. 3 (March), pp. 1157–1182, 2003.

[52] M. Claesen and B. De Moor, "Hyperparameter search in machine learning," *arXiv preprint arXiv:1502.02127*, 2015.

[53] P. Sedgwick, "Pearson's correlation coefficient," *Bmj*, vol. 345, 2012.

[54] F. Pedregosa *et al.*, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[55] T. Chen and C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785–794, ACM, 2016.

[56] F. Chollet *et al.*, "Keras: Deep learning library for theano and tensorflow." Available in: https://keras.io. Online. Accessed: Jun 12, 2020.

[57] J. Grus, *Data science from scratch: first principles with python*. O'Reilly Media, 2019.

[58] V. Buenestado, J. M. Ruiz-Aviles, M. Toril, S. Luna-Ramírez, and A. Mendo, "Analysis of throughput performance statistics for benchmarking lte networks," *IEEE Communications letters*, vol. 18, no. 9, pp. 1607–1610, 2014.

[59] C. Gijón, M. Toril, M. Solera, S. Luna-Ramírez, and L. R. Jiménez, "Encrypted Traffic Classification Based on Unsupervised Learning in Cellular Radio Access Networks," *IEEE Access*, vol. 8, pp. 167252–167263, 2020.

[60] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794, 2016.

[61] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.

[62] S. Ruder, "An overview of multi-task learning in deep neural networks," *arXiv preprint arXiv:1706.05098*, 2017.

**Carolina Gijón** received her B.Sc. degree in Telecommunication Systems Engineering and her M.Sc. Degree in Telecommunication Engineering from the University of Málaga, Spain, in 2016 and 2018, respectively. Currently, she is working towards the Ph.D. degree. Her research interests include self-organizing networks, machine learning and radio resource management.

**Matías Toril** received his M.S in Telecommunication Engineering and the Ph.D degrees from the University of Málaga, Spain, in 1995 and 2007 respectively. Since 1997, he is Lecturer in the Communications Engineering Department, University of Málaga, where he is currently Full Professor. He has co-authored more than 130 publications in leading conferences and journals and 8 patents owned by Nokia or Ericsson. His current research interests include self-organizing networks, radio resource management and data analytics.

**Salvador Luna-Ramírez** received his M.S in Telecommunication Engineering and the Ph.D degrees from the University of Málaga, Spain, in 2000 and 2010, respectively.Since 2000, he has been with the department of Communications Engineering, University of Málaga, where he is currently Associate Professor. His research interests include self-optimization of mobile radio access networks and radio resource management.