A deep map from performance to resources – intelligent resource allocation for network slicing

Afonso Oliveira* and Teresa Vazão*
*INESC-ID / Instituto Superior Técnico / Universidade de Lisboa

***Abstract*—Network slicing allows flexible and automatic provisioning of resources to satisfy contractually defined Key Performance Indicator (KPI) targets. When admitting new slices and during the slice run-time, the mapping between a KPI target and the resources required to fulfill it should be as accurate as possible. However, in mobile wireless networks, the number of resources to achieve a given network performance varies with the users' connection quality over time. An incorrect mapping may result in additional costs due to excessive over-provisioning or fines paid due to a contract breach. This work introduces a framework that predicts a cost efficient KPI target to resource mapping for a future time window. It does so by learning with network condition history and the slice's contractual agreement. Different operators can use the framework in both slice admission and run-time. Simulations show that the framework successively decreases costs related to an inefficient KPI target to resource mapping.**

***Index Terms*—network slicing, mobile wireless networks, resource allocation, neural networks**

I. Introduction

Network slicing comes as an enabler for the implementation of 5G and Beyond 5G systems by creating logical networks with traffic isolation between them [1]. This allows a single infrastructure to host distinct traffic types with different requirements while also introducing new possibilities for network sharing. Network slicing offers the possibility of contracting resources on one set of network infrastructures to achieve the Key Performance Indicator (KPI) targets of distinct traffic types, possibly through multiple hierarchically organized operators [2]. Slice management provides an allocation of physical resources based on one or more KPI targets that are part of a slice template, which performance can be secured by a Service Level Agreement (SLA) contract.

In the radio part of a mobile wireless network, the Radio Access Network (RAN), some works offer the possibility of allocating a specific number of radio resources over time [3]. These can be complemented by RAN slice schedulers [4] that ensure slice isolation at the network infrastructures. However, to satisfy a given KPI target the required radio resources to be allocated vary with the radio signal quality conditions of the affected User Equipments (UEs). This complicates the allocation process, since the varying radio propagation conditions affect future resource needs. An inadequate mapping between a KPI target and the resources required to achieve it may result in either too much resources or too few resources being allocated. The impact of this problem varies with the entity responsible for this mapping.

In a traditional scenario where a Mobile Network Operator (MNO) with direct access to the RAN is responsible for the mapping, there is a problem regarding slice admission. When resources are over-provisioned for the current slices, the admission of new slices is more limited. This results in the denial of new slices that could have been accepted, impacting the revenue. When resources are underestimated for a slice, its KPI target may not be reached during the slice lifetime. This may result in a fine due to the SLA contract violation.

In a multi-tenancy scenario, where a Mobile Virtual Network Operator (MVNO) allocates a specific number of resources to an Infrastructure Provider (InfP) to satify its clients' KPIs targets, there is a problem during slice management. The resource over-provisioning results in unnecessary resource acquisition costs, while the under-provisioning may also result in fines due to an SLA contract violation.

This introduces the need to design a flexible and efficient mapping between traffic performance targets and radio resources. This mapping should flexibly adapt itself to varying radio conditions, predicting future resource capacity values while figuring out the provisioning amount that is the most cost-efficient regarding costs related to over-provisioning and SLA contract violation fines.

One way to look for solutions are the RAN slice schedulers. However, while most of them are designed to be used with KPI targets, some of them state that the required mapping is out-of-scope [5][6]. The ones that specify some form of mapping use the present average of per-resource bitrate [7][8]. While it fits the scheduling algorithms' needs since they make scheduling decisions based in the past achieved performance, the mapping method could be improved when used in the solutions' slice admission procedure by using a prediction component.

The main influence in the information capacity of a radio resource is the connection quality of the slice's UEs. A number of works focus in the prediction of the radio connection quality [9][10]. However, these works focus on predicting the specific radio connection value. The desired mapping value for the problem we aim to solve should take into consideration the over-provisioning and contract violation trade-off so that it is cost-efficient for the operator that uses it.

Our contribution to solve the problem is Cost-aware Resource Prediction (CARP), a framework that uses neural networks to allow the prediction of a KPI target to resource mapping that aims to be cost-efficient for the operator using it. This cost-efficiency is obtained by exploring over-provisioning and SLA contract violation fines trade-off costs. The resulting mapping procedure can be used by RAN slicing algorithms and slice admission procedures when used by an MNO, or to provide a more efficient slice management to a MVNO that contracts resources to an InfP so that it can offer services to its clients based on KPI targets.

Section II surveys works that aim to predict radio signal quality. CARP is detailed in Section III. The framework is validated through simulation work in Section IV. Finally, Section V concludes the paper.

II. State of the art

This section describes works that learn with past radio quality traces to perform predictions of different nature. The selected works either apply machine learning to predict radio channel quality or any element affected by it such as the used Modulation and Coding Scheme (MCS).

Ardalani experimented Signal-to-Interference-plus-Noise Ratio (SINR) prediction with Adaline, a single layer linear neural network, and Multilayer Perceptron (MLP), a multi-layer neural network [11]. The setup aimed to predict the fading characteristics of the wireless signal to regulate transmission power in a closed-loop solution. More recently, Kulkarni *et al.* studied wireless signal quality of different wireless technologies using recurring neural networks: Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) [9]. These are compared with Autoregressive Integrated Moving Average (ARIMA) and a simple linear regression, with the deep learning techniques, LSTM and GRU, achieving better results. Also using deep networks, Luo *et al.* design a prediction system for the channel state information using a combination of Convolutional Neural Network (CNN) and LSTM [10].

As for learning techniques applied to the prediction of used MCS, Sayeed *et al.* uses the MCSs used over time with the number of scheduled Resource Blocks (RBs) to predict the amount of information scheduled at each scheduling opportunity [12]. The learning is done with ARIMA to improve adaptive streaming applications. In another application, Gutterman *et al.* combines information as the offered bitrate for each user (which depends on the used MCS) and its activity to predict the resource usage by active nodes [13]. This aims to help a network slice broker with slice admission decisions. Authors used both ARIMA and LSTM, as well as an introduced LSTM adaptation.

III. Cost-Aware Resource Prediction (CARP)

This section details this work's framework for providing a performance to resource mapping, CARP. CARP allows the mapping between a requested KPI target and the required resources to fulfill it. First, we explain how CARP can provide the mapping of KPI targets of distinct traffic groups, which justifies its input and output. Then, we detail CARP's learning process to adapt to the varying radio signal conditions.

A. Satisfying distinct traffic types

The main objective of CARP is to allow network operators to allocate the required radio resources to fulfill an SLA contract. SLA contracts often include KPI targets to be met and, in some scenarios, some pre-conditions such as traffic control admission are also stipulated. Due to traffic type diversity, the mapping between a requested KPI target and the required resources may not be achieved in a direct manner. From rate-base services, such as the enhanced Mobile Broadband (eMBB), to latency-sensitive services, such as Ultra Reliable Low Latency Communications (URLLC), the resource usage strategies to guarantee the KPI targets may vary.

For eMBB traffic, the target bitrate, which can be the KPI, can be sent to CARP to obtain the bandwidth in RBs required to achieve the target bitrate. For URLLC traffic, there could be a pre-condition such as an admission control procedure that defines the amount of traffic that is in-profile, and hence, that needs to be prioritized. The limitation in bytes of the admission control can be mapped to a limitation in resources. One example is the resource-based token bucket system in the work by Oliveira and Vazão [4]. Also Schmidt *et al.* [8] define a maximum amount of resources allocated at each scheduling opportunity, which can be mapped to a bitrate limitation.

In the studied cases we identified the average information capacity of each resource the core problem to solve. CARP will then receive as input an amount of information or bitrate required to achieve a given contract's KPI target or pre-condition. It will then return the radio resources required to transmit that amount of information. For instance, if the KPI target is a bitrate of 10Mbps, the network operator can inquire CARP what is the bandwidth in terms of RBs that can satisfy it. If a maximum packet size is stipulated for URLLC traffic, CARP can tell how many resources should be considered to accommodate that packet size. This volume of information that is the input of CARP is henceforth referred in this document as the KPI target, or simply target.

B. CARP learning

CARP has three main inputs for the learning process. They are the training set, which contains time-series information on the UEs, the cost trade-off, which is the trade-off between SLA violation and over-provisioning costs, and the KPI target, which is the target bitrate or information volume for which the mapping is desired. The output is a mapping from the target bitrate or volume of information to the resources required.

The output is achieved through predicting the ideal spectral efficiency for the future time window to serve the contract. This metric gives information on the achievable bitrate per frequency unit (bps/Hz). With this predicted value the number of resources, more specifically RBs, required to transmit or receive a given number of bytes or to achieve a given bitrate can be calculated.

1) Training set: For this work two separate groups of features are considered: minimalist and full. The minimalist group includes only Reference Signal Receive Power (RSRP) and Channel Quality Index (CQI). These are provided by the

UE through periodic measurement reports and channel-state reports. The full group of features include each UE's position, speed, RSRP, Reference Signal Received Quality (RSRQ), Signal-to-Noise Ratio (SNR), and CQI. This latter group is to be used in a scenario where the UE reports all these values periodically through means outside of Long Term Evolution (LTE) or New Radio (NR) scopes.

RSRP, RSRQ, SNR are different signal power and quality metrics collected by the UE regarding the downlink direction. CQI is an index that represents the MCS being used, from which the spectral efficiency can be extracted. For the minimalist group, the chosen signal quality metric is RSRP since it is communicated to the eNodeBs (eNBs) by default [14], as opposed to SNR, and does not depend on the number of RBs used for measuring, as opposed to RSRQ.

The value of spectral efficiency used for comparing the neural network prediction with the real value is extracted from CQI. This is done using a table from the LTE or NR standards that maps the CQI, to the MCS used which dictates the achieved spectral efficiency [15].

2) Neural networks: For this work we chose to use neural networks since they allow the use of custom loss functions and multivariate regression. We use the LSTM model, since our aim is to predict a time series. LSTM is a neural network that allows the exploration of temporal relationship between different entries. This model is used in previous works that do signal quality prediction in wireless mobile networks [9][10].

3) Prediction: By using the information in the training set, the neural networks give as output a spectral efficiency value. This value is not a typical prediction trying to get a value as close to the real outputs in the training set as possible. For that a symmetric loss function is typically used.

CARP uses a custom asymmetric loss function to predict the spectral efficiency in a future time window that avoids an SLA contract violation. Since a small over-provisioning is preferred to a contract violation, the error between the real and predicted value should not be symmetric.

Contract related costs and the KPI target to achieve are two more inputs that are important to the learning process despite not being used as training features. These are used in the loss function so that the predictions take into account costs related to over-provisioning and contract breach, or SLA violation. The target also affects the loss function, since it affects the amount of required resources, that affect the over-provisioning and contract breach costs of a spectral efficiency prediction error.

As referred the target value considered can be either a bitrate or a volume of information. This $target$ is also used after the spectral efficiency prediction (η) has been done to calculate the amount of required resources (N_{RB}) following Equation (1). In the frequency component, an RB is composed by 12 sub-carriers each with a Δf spacing, 15 KHz bandwidth in LTE. By multiplying by the spectral efficiency value, the bitrate of one RB is obtained. The target bitrate over one RB's bitrate gives the bandwidth in RBs required to serve the given bandwidth. If a volume of information is the target, then a division by the duration of one RB (T_{RB}), 0.5 ms in LTE, is required[1].

$$N_{RB}(target, \eta) = \begin{cases} \left\lceil \dfrac{target}{12 \times \Delta f \times \eta} \right\rceil, & \text{if bitrate} \\ \left\lceil \dfrac{1}{T_{RB}} \dfrac{target}{12 \times \Delta f \times \eta} \right\rceil, & \text{if volume} \end{cases} \tag{1}$$

CARP's output is the mapping between a performance target and the RBs required to achieve it over a future time window. For that it uses the predicted spectral efficiency and the input performance along with Equation (1).

4) Asymmetric loss function: So the learning models consider the SLA violation/over-provisioning trade-off, the α-OMC [16] loss function proposed by Bega *et al.* is adapted. This loss function gives a fixed penalty to under-estimations and a linear penalty to over-estimations. The α in α-OMC is the amount of additional resources that gives an over-provisioning cost equal to the SLA violation fine.

In the context of this work, we also consider the over-provisioning cost to increase linearly with the additional resources allocated beyond the required amount. However, the relationship between resources and the predicted value, spectral efficiency, is not linear as observed in Equation (1). The adaptation of α-OMC to this work, η-OMC, calculates the resource difference, d, required to satisfy a $target$ from the predicted, $\hat{\eta}$, and real, η, spectral efficiency values using Equation (2). For training, the values of N_{RB} are used without the ceiling operator.

$$d(target, \hat{\eta}, \eta) = N_{RB}(target, \hat{\eta}) - N_{RB}(target, \eta) \tag{2}$$

The definition of η-OMC is represented in Equation (3), where t is the target performance, α the SLA violation over the over-provisioning cost, and ε a small number to assure the neural network convergence [16]. The shape of the loss function with different target values can be seen in Figure 1. It can be observed that for the same trade-off, the higher the target, the more difficult is the cost minimization since there is a low error possible to avoid costs comparable to the contract violation cost.

$$\eta\text{-OMC}(\alpha, \mathrm{t}, \hat{\eta}, \eta) = \begin{cases} 1 - \dfrac{\varepsilon\ d(t, \hat{\eta}, \eta)}{\alpha}, & \text{if } d(t, \hat{\eta}, \eta) \le 0 \\ 1 - \dfrac{d(t, \hat{\eta}, \eta)}{\varepsilon}, & \text{if } d(t, \hat{\eta}, \eta) > 0 \land \\ & \quad d(t, \hat{\eta}, \eta) \le \varepsilon \\ \dfrac{d(t, \hat{\eta}, \eta) - \varepsilon}{\varepsilon + \alpha}, & \text{if } d(t, \hat{\eta}, \eta) > \varepsilon \end{cases} \tag{3}$$

[1]The values of subcarrier spacing and RB/slot duration can be changed to adapt to the different New Radio numerologies for 5G implementations.

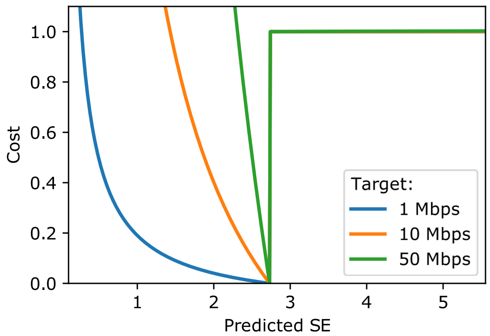

Figure 1. η-OMC loss function. $\eta = 2.7305$, $\alpha = 50$.

IV. Simulation work

The proposed solution was trained and validated recurring to a dataset with real traces. This section describes how the dataset was manipulated and the models were trained, what scenarios were simulated, and the results of simulations with a validation subset of the dataset. The used implementation was made publicly available [17].

A. Pre-processing

This section introduces the dataset used and explains how it was processed for the learning process as well as model validation. Following that the neural networks' configurations and parameterization are presented.

1) Dataset and data preparation: In order to train and validate the algorithm, real traces from LTE network were used. The used UE traces were made available by Raca *et al.* [18] and include samples with the node position, channel measurements, and the indicated CQI with a time granularity of 1s. Each trace includes as well, the cell the UE is associated to. The spectral efficiency value is obtained from the CQI value using a table from the LTE standard [15]. The UE traces have variable sizes.

In the simulation scenario, we consider a time window of 10s and the models are trained with information from a 20s window, which results in a total of 30s of data required per training entry: with 20s of data we predict the resources required for the next 10s. Considering the duration of the available traces in the dataset, these values were found to be a good compromise so that there is a reasonable amount of entries for the neural networks.

All the traces were separated into chunks, each chunk being characterized by the cell the UE is connected to. Only the values of cell 2 were considered since that cell includes considerably more samples than the others. After the split of traces in chunks by cell, the dataset was split into training, testing and validation based on a ratio of 80%, 10%, 10%, respectively. The ratio is applied to the total number of samples of all traces, however individual traces are not separated. The traces were then divided in 30s windows. These are sliding windows separated by 2s, meaning each two consecutive divisions have 28s of data in common. For instance, considering w_n one of this windows, $w_1 = [0 : 30[$, $w_2 = [2 : 32[$, $w_3 = [4 : 34[$, and so on. This data augmentation process was done to increase the number of inputs for training.

A Full and Minimum training sets are extracted as referred in Section III-B1. The features of each of these sets is listed in Table I. The spectral efficiency (SE) is obtained through table lookup, from [15], of the registered CQI value.

Table I
Training sets

Training set	Features
Full	Latitude, Longitude, Speed, RSRP, RSRQ, SNR, SE(CQI)
Minimum	RSRP, SE(CQI)

2) Neural network configuration: The neural network has been configured to use 2 hidden layers and 30 units per layer. All layers have a dropout rate of 0.1 to avoid over-fitting. This parameterization was tuned based on experimental results with different parameters using the training set and the test dataset for result comparison. For the implementation of the learning process we used Keras framework.

B. Simulation setup

This section describes the simulation setup used. First the alternative mapping strategies to CARP are presented, then it is described how the dataset is used to compare the performance between solutions, and finally the simulation scenarios are listed.

1) Alternative mapping strategies: For comparative purposes, the machine learning solutions were compared with two static strategies: one where the last spectral efficiency value is used to make the next 10s reservation (*Same*), and another more conservative where the spectral efficiency is considered be the one that maps to the CQI directly below the last used value (*Same-1*). For instance, if the last CQI value used was 13, then the spectral efficiency value to use is the one that is obtained from a CQI value of 12. Additionally, the CARP learning solutions are also compared with neural network solutions with the same features, but using a symmetric loss function: Mean Squared Error (MSE).

2) Model validation process: The performance of the models used for training and the static strategies is obtained by calculating the resource reservation cost using the validation dataset. For each 30s window, the models and static strategies predict the spectral efficiency for the next 10s window with the last 20s of data. From this value, the number of RBs to be requested for the next 10s to provide the target bitrate is calculated using Equation (1).

For each CQI value in the next 10s window, the required number of RBs is also calculated using the respective spectral efficiency value of the CQI table [15]. The number of requested and required RBs are then compared at each second to check if there is a contract violation, or how many RBs were requested above the required value.

The SLA violation cost is 1 and the over-provisioning is $1/\alpha$ per additional requested RB beyond the required. This is equivalent to the η-OMC cost function with $\varepsilon = 0$.

3) Simulation scenarios: Simulations are run using different relationships between violation and over-provisioning costs with α being the number of additional RB which the over-provisioning cost equals the contract violation cost: the values of 20, 50, and 100 are used. The targets considered for the work's validation are bitrate targets for each UE. The considered values vary between 1, 10, and 50 Mbps. These combinations were selected as they cover both complicated scenarios, where the cost minimization is difficult to achieve resulting in contract violations to avoid excessive over-provisioning costs; and simple scenarios, where a lower cost can be easily achieved without the existence of contract violations.

C. Results and analysis

This section presents the results and analysis of the simulated scenarios.

1) Cost performance comparison: First, we compare different solutions in the different scenarios: using the Full training set and η-OMC as a loss function, using the Minimum training set and η-OMC as a loss function, using the Full training set and a symmetric loss function (MSE), and the two static solutions *Same* and *Same-1*. The aim is to evaluate how CARP fares with the different training sets and against other strategies. Figure 2 illustrates the results.

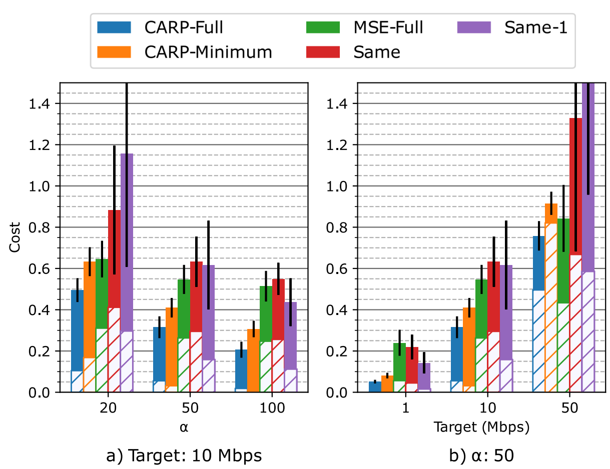

Figure 2. Strategy comparison. Dashed bars illustrate the percentage of SLA violations.

Figure 2a) varies α using a fix target bitrate at 10 Mbps, while Figure 2b) compares different target values with a fix α of 50. Dashed bars illustrate the percentage of SLA violations, where 100% is equivalent to the height of the fully colorized bar. Each bar corresponds to the average value of the 30s windows obtained from the validation dataset with the black lines indicating the 95% confidence interval.

Remember that by varying the target bitrate, the available predicted values to achieve a low cost (lower than the violation fine) also varies. The same happens by varying α. Lower $alphas$ and higher targets represent scenarios where it is more difficult to achieve a low cost. In the most difficult scenarios represented, $alpha = 20$ is plot a) and $Target = 50Mbps$ in plot b), the results are higher similarity between the features and loss function used. In this scenario, the static strategies incur in high costs. Note that this is not a probable scenario in real life. The high number of SLA violations in the solutions with an asymmetric loss function (Full and Minimum) while having lower costs, reveal that risking SLA violations is better than doing more over-provisioning. If that were the case, then there would be no interest in making the SLA contract in the first place. However, studying this extreme scenarios helps to understand CARP behavior.

As the scenarios have higher availability for lower costs, that is going from left to right in the a) plot, or the opposite in the b) plot, the static strategies begin to have better performance than the symmetric cost (MSE) approach. This demonstrates the unsuitability of the MSE loss function for this problem. The approaches using CARP (Full and Minimum), end up achieving lower cost values than the remaining. From the both approaches using CARP, the full feature set achieves better results overall. It should be noted that despite the solution using the full feature set having better performance than the one using the minimum set, the obtained cost difference is low. Also, the approach with a minimum feature set still outperforms the remaining solutions.

2) Applying the same model to different KPI targets: One limitation of CARP is the requirement of a fix KPI target for training each model. Separate models have to be trained when different KPI targets are to be considered. However, the target for the loss function and the target used for the mapping procedure can vary. CARP can be trained with a carefully chosen finite set of key targets to get a model for each of those targets, T_k. When using CARP in a real scenario with a target t as input, the model trained with the closest key target T_k can be used to get the predicted spectral efficiency and the number of resources calculated using $target = t$. To test this, we analyzed the performance of one fixed model trained with a target of 10Mbps applied to different targets. The resulting cost is compared to the models that are trained with the specific target CARP is being used with.

Figure 3 shows the result of this comparison. The target values from 5 to 15Mbps are presented along with the values of 20, 30, 40, and 50Mbps. Fixed10 represents the results of using CARP with a model trained with a target of 10Mbps, while Adaptive represents the results of using CARP with a model trained with the target being used. The Fixed10 model achieves a similar performance when comparing to the Adaptive models in the target values around 10Mbps, more specifically 6 to 15Mbps. With a 5Mbps target, the Adaptive model obtains a lower cost. Regarding the tests with 20 and 30Mbps, some variations between the used models exist, but each result is within the confidence interval of each other. With 30Mbps the result is the opposite of what is expected, probably due to specific entries in the validation dataset since the confidence interval is relatively high. With the target values of 40 and 50Mbps the Adaptive models have a clear advantage, as expected. These results show that the proposed solution of training with a finite number of targets T_k is viable.

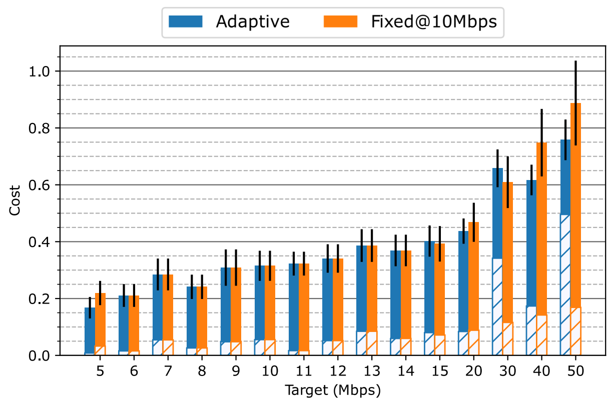

Figure 3. Adaptive models and fixed model at 10Mbps with $\alpha = 50$.

V. Conclusions

The main focus of this work is the minimization of operational costs through the consideration of contract violation and over-provisioning associated costs. For that we introduce CARP, a framework that uses neural networks to aid in the process of resource allocation that allows mapping KPI targets to the required physical radio resources.

CARP was compared to a similar solution using a symmetric loss function and two static strategies: one simple and another more conservative. Overall, results demonstrated a good performance from learning models used under a different set of scenarios where the balance between contract violation and over-provisioning as well as targeted performance varied.

CARP was also tested using the access to a full and minimum feature set. The minimum feature set used information obtained as part of the LTE and NR standard channel updates, while the full feature set had additional signal quality metrics and node position. The validation with the full feature set resulted in slightly lower costs, however the difference between both approaches was small and below the alternatives. This shows that CARP does not depend on information not already available in current base stations of mobile wireless networks, avoiding the collection of sensible information.

As a future work, UE traffic activity could also be considered. If the UE's with the lowest signal quality are responsible for most of the exchanged information, then the overall slice performance to resource mapping will be different. Also, with access to datasets with specifics on traffic scheduling, the amount of link layer re-transmissions can be taken into account for a more precise KPI target to resource mapping at higher layers.

Acknowledgments

This work was supported by national funds through Fundação para a Ciência e a Tecnologia (FCT) with reference UIDB/50021/2020 and SFRH/BD/132053/2017.

References

[1] Xenofon Foukas, Georgios Patounas, Ahmed Elmokashfi, and Mahesh K Marina. Network slicing in 5g: Survey and challenges. *IEEE Communications Magazine*, 55(5):94–100, 2017.

[2] K. Samdanis, X. Costa-Perez, and V. Sciancalepore. From network sharing to multi-tenancy: The 5g network slice broker. *IEEE Communications Magazine*, 54(7):32–39, 2016.

[3] Xenofon Foukas, Navid Nikaein, Mohamed M. Kassem, Mahesh K. Marina, and Kimon Kontovasilis. Flexran: A flexible and programmable platform for software-defined radio access networks. In *Proceedings of the 12th International on Conference on Emerging Networking EXperiments and Technologies*, CoNEXT '16, pages 427–441, New York, NY, USA, 2016. ACM.

[4] Afonso Oliveira and Teresa Vazão. Adapting priority schemes to achieve network slice isolation. In *Proceedings of the 35th Annual ACM Symposium on Applied Computing*, SAC '20, page 1164–1171, New York, NY, USA, 2020. Association for Computing Machinery.

[5] A. Ksentini and N. Nikaein. Toward enforcing network slicing on ran: Flexibility and resources abstraction. *IEEE Communications Magazine*, 55(6):102–108, June 2017.

[6] T. Guo and A. Suárez. Enabling 5g ran slicing with edf slice scheduling. *IEEE Transactions on Vehicular Technology*, 68(3):2865–2877, March 2019.

[7] Ravi Kokku, Rajesh Mahindra, Honghai Zhang, and Sampath Rangarajan. Nvs: A virtualization substrate for wimax networks. In *Proceedings of the Sixteenth Annual International Conference on Mobile Computing and Networking*, MobiCom '10, pages 233–244, New York, NY, USA, 2010. ACM.

[8] R. Schmidt, C. Chang, and N. Nikaein. Slice scheduling with qos-guarantee towards 5g. In *2019 IEEE Global Communications Conference (GLOBECOM)*, pages 1–7, 2019.

[9] Adita Kulkarni, Anand Seetharam, Arti Ramesh, and J Dinal Herath. Deepchannel: Wireless channel quality prediction using deep learning. *IEEE Transactions on Vehicular Technology*, 69(1):443–456, 2019.

[10] C. Luo, J. Ji, Q. Wang, X. Chen, and P. Li. Channel state information prediction for 5g wireless communications: A deep learning approach. *IEEE Transactions on Network Science and Engineering*, 7(1):227–236, 2020.

[11] Nahid Ardalani. Sinr prediction in mobile cdma systems by linear and nonlinear artificial neural-network-based predictors. *International Journal of Computer Science Issues (IJCSI)*, 8(4):220, 2011.

[12] Z. Sayeed, Q. Liao, D. Faucher, E. Grinshpun, and S. Sharma. Cloud analytics for wireless metric prediction - framework and performance. In *2015 IEEE 8th International Conference on Cloud Computing*, pages 995–998, 2015.

[13] Craig Gutterman, Edward Grinshpun, Sameer Sharma, and Gil Zussman. Ran resource usage prediction for a 5g slice broker. In *Proceedings of the Twentieth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Mobihoc '19, page 231–240, New York, NY, USA, 2019. Association for Computing Machinery.

[14] Erik Dahlman, Stefan Parkvall, and Johan Skold. *4G: LTE/LTE-Advanced for Mobile Broadband*. Academic Press, Inc., Orlando, FL, USA, 2nd edition, 2013.

[15] 3GPP. Evolved Universal Terrestrial Radio Access (E-UTRA); Physical layer procedures. Technical Specification (TS) 36.213, 3rd Generation Partnership Project (3GPP), April 2017. Version 14.2.0. Table 7.2.3-1.

[16] D. Bega, M. Gramaglia, M. Fiore, A. Banchs, and X. Costa-Perez. α-omc: Cost-aware deep learning for mobile network resource orchestration. In *IEEE INFOCOM 2019 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 423–428, 2019.

[17] Afonso Oliveira and Teresa Vazão. A deep map from performance to resources (scripts and models), October 2021. Zenodo. Version v0.1.0. https://doi.org/10.5281/zenodo.5203250.

[18] Darijo Raca, Jason J. Quinlan, Ahmed H. Zahran, and Cormac J. Sreenan. Beyond throughput: A 4g lte dataset with channel and context metrics. In *Proceedings of the 9th ACM Multimedia Systems Conference*, MMSys '18, page 460–465, New York, NY, USA, 2018. Association for Computing Machinery.