Uncertainty-aware RAN Slicing via Machine Learning Predictions in Next-Generation Networks

Raouf Abozariba[1], Muhammad Kamran Naeem[2], Md Asaduzzaman[3] and Mohammad Patwary[1]

[1]School of Computing and Digital Technologies, Birmingham City University, UK
[2]Research Innovation and Enterprise, Solent University, UK
[3]School of Creative Arts and Engineering, Staffordshire University, UK
{r.abozariba, kamran.naeem, asad, m.n.patwary}@ieee.org

***Abstract*—Network slicing enables 5G network operators to offer diverse services in the form of end-to-end isolated slices, over shared physical infrastructure. Wireless service providers are facing the need to plan and rapidly evolve their slices configuration to meet the varied tenants' demand. Network slicing unfolds a new market dimension to the infrastructure providers as well as to the tenants, who may acquire a network slice from the infrastructure provider to deliver a specific service to their respective subscribers. In this new context, there is a growing need for new network resource allocation algorithms to capture such proposition. This paper addresses this problem by introducing a family of online algorithms with the aim to (i) minimize tenants spectrum allocation costs, (ii) maximize radio resource utilization and (iii) ensure that the service level agreements (SLAs) provided to tenants are satisfied. We focus on improving the performance of prediction-based decisions that are made by a tenant when prediction models lack the desired accuracy. Our evaluations show that the proposed probabilistic approach can automatically adapt to prediction error variance, while largely improving network slice acquisition cost and resource utilization.**

***Index Terms*—RAN Slicing, spectrum management, 5G, traffic forecasting, machine learning, next generation wireless networks.**

I. Introduction

Worldwide mobile data traffic is set to increase at a rate of 25% and 42% by the year 2025 [1]. This extreme explosion, driven by emerging applications, opens new business horizons and models, and brings vertical segments into the networking industry. The need for high throughput demand, low latency as well as inclusion of new verticals has been met by the recent 3GPP releases of fifth-generation (5G) networking specifications, which offer speeds of 20 Gbps or higher and have latency in the order of milliseconds. Among the core features included in various 3GPP study items in the context of 5G cellular network, lies in the flexibility of network architecture, facilitated by two critical pillars: network virtualization and network slicing. These technologies may also be included in fourth-generation (4G) as part of the (LTE)-Advanced Pro standard.

According to various sources, 5G network slicing is an approach to provide separate independent end-to-end (E2E) logical network resources to serve applications with various demands [2]. Those applications range from autonomous vehicles to various machine-to-machine communications, providing services such as enhanced mobile broadband (eMBB), massive machine-type communications (mMTC), and ultra-reliable and low-latency communications (URLLC) [3]. An E2E network slice consists of radio access network (RAN) slice, core slice and transport slice, each with their own controller, already defined in recent 3GPP technical specification releases. Fig. 1 illustrates typical E2E network slicing concept, a simplified version of the one found in [4] and other studies. With hundreds of varied slices available, it is also possible for mobile network operators (MNO) to determine what slice types each customer (also known as tenant) is eligible to use based on defined service level agreement (SLA) and subscriptions [5]. A slice can be created to give service to a specific group of devices with the same requirements (sensors or smartphones) or by type of application (e.g., a slice for multimedia services). There are few network equipment manufacturers offering different forms of slicing capabilities, although these available options of E2E network slicing remains in the development phase and largely still under investigation. The slicing technology is expected to dominate future network access mechanisms, both in the core and at the edge of the network.

RAN slicing, an important segment of the E2E slicing, has recently gained significant popularity, both from academia and industry, and has the potential to provide cost-effective solutions for network management, as discussed in the literature [6]–[8]. It has been reported in [9] that RAN slicing can save billions of USD in capital expenditure and operational expenditure by 2025 worldwide. RAN slices are implemented at sites by arranging the radio resources into a number of carriers and assigning them to each tenant, enabling operators to develop intelligent scheduling and to mitigate interference. Static splitting of radio resources can be inefficient under variable load, leading to the development of new RAN slicing architectures that enable dynamic resource sharing. Those include vRAN, FlexRAN and openRAN [10], [11]. A common goal among these architectures is to reduce resource costs and satisfy SLAs requirements to tenants. However, it is challenging for schedulers at the edge of the network to know how long a certain job will run and what demand might arrive in the future. It is therefore expected that RAN slicing will be traffic-aware to enable allocation according to a tenant's needs. Moreover, it is envisaged that the RAN will permit tenants to access multiple slices simultaneously

through carrier aggregation [4].

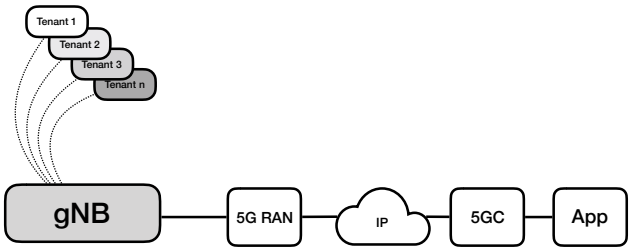

Fig. 1: Main blocks of end-to-end network slicing

In order for dynamic slice allocation to be reasonably effective, service providers must have an accurate view of the state of traffic conditions on the network and be able to predict, at least over short periods, how the current network conditions will evolve over time. Generally, prediction accuracy can vary widely due to noise that is often inherent in forecasts and cannot always be taken as gospel.

In machine learning community there are many efforts to improve the overall accuracy of predictors, measured as an error on the testing data. Machine learning techniques have been applied to multiple areas in mobile networks, including radio access technology (RAT) and mobile traffic prediction. Prediction techniques such as spatio-temporal neural network (STN), auto-regressive integrated moving average (ARIMA) and Holt-Winters exponential smoothing (HW-ExpS) provide varying degrees of accuracy, when applied to traffic prediction [12], [13].

Such clustering and regression techniques rely on the available data on which the prediction of users' demand is based. In cellular networks, such data can be obtained using Charging Data Records (CDRs), which contain information such as time and amount of data a user uploads and downloads, as well as the identity of the associated base station [14]. However, data from CDRs are sparse in time as they are obtained only when users engage in data transmission and their location is recorded only at the granularity of a base station [14], [15]. Moreover, when new networks are deployed, training data is often unavailable or do not reflect the desired test distribution. Another limitation of available prediction solutions is the inability to cope with the increased challenges presented by the unpredictable congestion in physical layer and dynamic environments [15]. Prediction models based on such data may produce wild variations in traffic estimation, providing unreliable predictions, which may lead to severe consequences, for example, under-forecasting causes SLA violations, while over-forecasting incurs unnecessary costs to the tenants and low resource utilization. SLA violations result in substantial penalties to the operator, therefore, it is important to design allocation strategies taking into account such SLA requirements.

Techniques based on offline training procedures used for mobile traffic forecasting will play a major role in defining the success of network slicing. Operators will continue to rely on prediction accuracy to create network slices to cope with future demands. Based on predictions, the operators may decide to allocate various slices with a varying number of resource blocks to tenants for either short periods of time on slot-by-slot basis or reserve a slice for a certain amount of time.

Contribution: In light of the above discussion, the challenge is to provide a cost effective decision making process of resource allocation to network slices. In this paper, we discuss the applicability of online algorithms to solve problems associated with resource allocation prediction in the presence of error variance. We show that it is possible to minimize the impact of unpredictable slice traffic load, providing lower costs to tenants under guaranteed SLAs. We focus our work only on guaranteed SLAs service as best-effort communication is unacceptable for many recent applications.

The rest of the paper is organized as follows. In Section II, we model the problem. The proposed baseline and probabilistic approaches, solutions and algorithms are given in Section III and IV. In Section V we report our findings, which were obtained through extensive simulations. Finally, we draw our conclusions and outline future works in Section VI.

II. Modeling the Problem

Before we introduce the system, we give an example (as shown in Fig. 2) of predicted traffic and actual demand, which will aid in understanding the rationale behind our work. The horizontal dashed lines divide the resource blocks into slices. As can be seen in the plot, the number of required slots does not always match with the predicted demand. One of the goals of the traffic prediction task is to minimize the allocation cost over the billing cycle while complying with user plane SLA requirements [16]. Let us consider a generic model, where a mobile network operator (MNO), who owns radio resources, provides RAN slices to tenants.

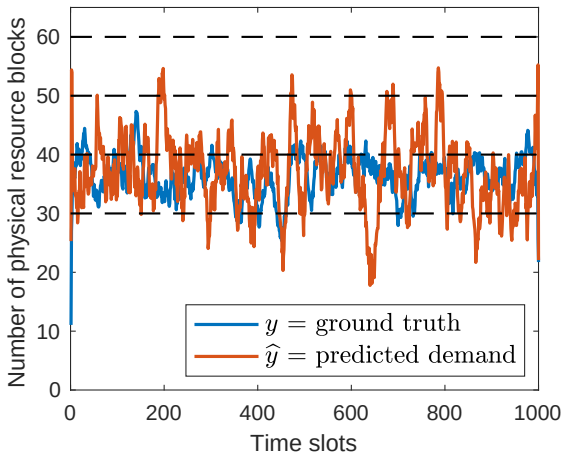

Fig. 2: Example of predicted demand

To meet the expected service requirements, the MNO manages a resource pool and allocates resource blocks to slices according to tenants predicted demand. Now, suppose that an MNO creates slices $\mathcal{S} = \{s_1, s_2, \ldots, s_N\}$, each provide a number of resource blocks. We assume each slice is divided into $\mathcal{T} = \{1, 2, \ldots, T\}$ time slots of uniform length. An example of radio resource demand over T time slots and N

slices is shown in Fig. 3. At each time slot, tenants may issue slice requests to the MNO according to tenant's real-time demand. The tenant may also issue a request to the MNO to buy all the slots of a particular slice at time 0 or at time $T - t$. The tenants determine their allocation strategy based on a certain prediction model. In contrast to offline prediction, the online prediction has limited duties and makes only certain decisions in certain situations, hence they can be run in polynomial time.

To give the reader a better understanding of the model, we define:

- y: as the demand (total number of slots out of T slots available per slice a tenant requires), which is unknown to the tenant during the predicting phase and only learns it at end of the slice time window T.
- $\widehat{y}$: as the estimated demand, predicted by a given ML model.
- $\{u_i\}$: as the cost per slot of the ith slice, used when considering allocation on demand.
- $r_i \in \{r_1, \ldots, r_N\}$: as the fixed cost a tenant has to pay for acquiring slots for a network slice at any time t up to the end of the slice duration T.
- $\delta = |y - \widehat{y}|$: as the prediction model error defined as the absolute difference between prediction and the actual demand, assumed to be Gaussian, independent and identically distributed with mean zero and finite variance.

If $\delta \approx 0$, the predictable part of the process is dominant; on the other hand if $\delta \gg 0$, the unpredictable part of the process is dominant. The optimum cost in this case is $\mathcal{A} = \min\{uy, r\}$, where $\mathcal{A}$ denotes the cost of a solution obtained by a given algorithm. The estimator's goal is to minimize the total number of prediction errors over the billing cycle. Next, we propose a baseline approach to choose between buying and renting.

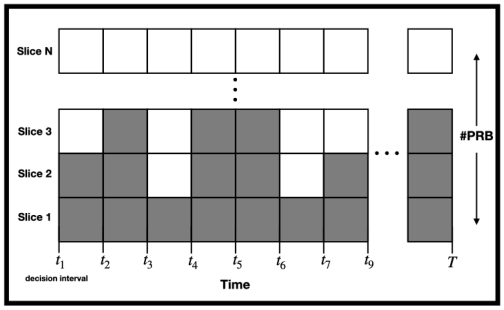

Fig. 3: An illustration of radio resources demand over T slots and N slices.

III. Baseline Approach

Problem formulation and approach: We now develop the problem through a mathematical optimization approach. As a tenant chooses a strategy; to buy or rent radio resources time slots by minimizing the cost, we can formulate the optimization problem by the following stochastic programming problem. Let x_i be a binary decision variable indicating whether to buy the kth slot at t_0 or rent per slot to the end of slice length T. The cost minimization problem can be written as

$$(\mathcal{P}_1) \quad \min \quad \sum_{i=1}^{N} \mathbb{E}_{\mathcal{D}} \left[Q(x_i, \mathcal{D}) \right] \qquad \text{s.t.} \quad x_i \in \{0,1\} \quad \forall i \in N \tag{1}$$

where

$$\begin{aligned} Q(x_i, \mathcal{D}) &= \min_i \quad [\, u_i \, \widehat{y} \,] \\ &= \min_i \quad \left[u_i \, (y \cdot \widehat{f}(y)) \right] \\ &\text{s.t.} \quad u_i \, \widehat{y} \geq r \end{aligned} \tag{2}$$

and $\widehat{y} = \sum y \, \widehat{f}(y)$. The optimization problem $(\mathcal{P}_1)$ requires the empirical distribution of the demand $\widehat{f}(y)$ to be specified. Therefore, the decision strategy equals to

$$\sum_{i=1}^{N} x_i = \begin{cases} \text{Buy}, & \widehat{y}\, u_i \geq r \\ \text{Rent}, & \widehat{y}\, u_i < r. \end{cases} \tag{3}$$

The optimization problem $\mathcal{P}_1$ is classical Ski Rental problem [17], with a trade-off between buying and renting when the number of slots is not known in advance to the tenant. $\mathcal{P}_1$ can be solved using Algorithm 1 which is described next.

Algorithm Description: Algorithm 1 (we also refer to this algorithm as a baseline approach) is an online algorithm with the following strategy: if the cost of the predicted number of slots is greater than the cost to reserve the entire available slots, then the algorithm reserves the slice up to T. If the cost of the predicted number of required slots is less than the cost to reserve the entire available slots, then the algorithm reserves the slice based on real-time demand per slot at cost u_i. At the end of the billing period, the algorithm also reveals a cost function, $\mathcal{A}$.

Algorithm 1: Baseline

Input: Let $\widehat{\mathbf{y}}$ be predictions of $\mathbf{y}$, $\mathbf{r}$ is the cost of buying the entire slice and $\mathbf{u}$ is the cost per slot.
Output: $\mathcal{A}$.
for $j = 1, 2, \ldots, |\mathcal{S}|$ **do**
  while $T \neq 0$ **do**
    if $\widehat{y}_j \, u_j \geq r_j$ **then**
      $a_j = r_j$
    else
      $a_j = u * \sum_{l=1}^{N} \mathbf{1} \cdot (d_{t_l} > 0)$, where d_{t_l} is the real-time demand per slot and $\mathbf{1}$ is the indicator function.
  $\mathcal{A} = \mathcal{A} + a_j$

Algorithm Analysis: To evaluate how well the baseline algorithm performs, we use the competitive ratio which is a function of the predictor's error δ as in [18]. We show that the algorithm performance can be categorized by $[\mathcal{A} \leq \mathcal{G} + \delta]$,

where $\mathcal{G}$ represents the optimal solution and $\mathcal{A}$ as defined previously.

$$\mathcal{A} = \begin{cases} \mathcal{G} = r, & \text{if } [\widehat{y} \cdot u \geq (r)] \text{ and } [y \cdot u \geq (r)] \\ \mathcal{G} = y \cdot u, & \text{if } [\widehat{y} \cdot u < (r)] \text{ and } [y \cdot u < (r)] \\ \mathcal{G} + \delta, & \text{if } [\widehat{y} \cdot u \geq (r)] \text{ and } [y \cdot u < (r)] \\ \mathcal{G} + \delta, & \text{if } [\widehat{y} \cdot u < (r)] \text{ and } [y \cdot u \geq (r)] \end{cases}$$

The first case means that the prediction $\widehat{y}$ and observation y are both higher than the cost r, which implies that the algorithm produces the optimal solution with $\delta = 0$. The second case means that the prediction $\widehat{y}$ and observation y are both lower than r, which implies that the algorithm again produces the optimal solution. The third case is derived as follow:

$$\begin{aligned} \mathcal{A} &= r \\ r &\leq y \cdot u + \underbrace{\widehat{y} \cdot u - y \cdot u}_{\delta} \\ \mathcal{A} &= y \cdot u + \delta \\ \mathcal{A} &= \mathcal{G} + \delta. \end{aligned}$$

Similarly, the fourth case is derived as:

$$\begin{aligned} \mathcal{A} &= y \cdot u \\ \widehat{y} \cdot u &< r \\ 0 &< r - y \cdot u \\ y \cdot u - y \cdot u &< r - \widehat{y} \cdot u \\ y \cdot u &< r + \underbrace{y \cdot u - \widehat{y} \cdot u}_{\delta} \\ \mathcal{A} &= r + \delta \\ \mathcal{A} &= \mathcal{G} + \delta. \end{aligned}$$

The analysis of algorithm 1 shows that its performance degrades with the error of the predictor and it is possible that the solution can move away from the optimal solution. More specifically, if $\widehat{y}$ is small and y is greater than r, δ could be large, resulting in either the high cost to the tenant or a degraded performance to its corresponding subscribers, violating SLAs requirements. In this case, the baseline algorithm presented here may not be the best choice for deciding between buy and rent in RAN slicing solutions. More analysis will be provided in Section V.

IV. Probabilistic Approach under imperfect look-ahead

Problem formulation and approach: To overcome the shortcomings of the previous program, we develop here an optimization problem, considering the uncertainty of number of slots. Let x_i be the decision variable whether to buy after renting j number of slots. The problem can now be formulated as

$$\begin{aligned} (\mathcal{P}_2) \quad \min \quad & \sum_i \mathbb{E}\left[Q(x_i, \mathcal{P}, \mathcal{D})\right] \\ \text{s.t.} \quad & x_i \in \{0,1\}, \end{aligned} \tag{4}$$

where

$$\begin{aligned} Q(x_i, \mathcal{P}, \mathcal{D}) &= \min_i \quad [\, u_i \, \widehat{y} \,] \\ &= \min_i \quad \left[u_i \left(\sum_{j=1}^{n} j \cdot p_j \right) \right] \\ &\text{s.t.} \quad u_i \, \widehat{y} \geq r, \end{aligned}$$

and the probability p_j is the distributions of renting slots before reserving the remaining part of the slots, which are given by

$$p_j = \left(\frac{r-1}{r} \right)^{k-j} \cdot \frac{1}{r\left(1 - \left(1 - \frac{1}{r}\right)^{k}\right)}, \quad j = 1, 2, \ldots, n \tag{5}$$

where $k = \lfloor \lambda \, r \rfloor$ if $\widehat{y} u_i \geq r_i$ and $k = \lceil b/\lambda \rceil$ if $\widehat{y} u_i < r_i$. $\lambda \in (0,1)$ is a hyper-parameter. The $\widehat{y}$ is the expected number of slots based on the empirical distribution $\widehat{f}(y)$. $\mathcal{P}_2$ is a variant of the Ski Rental where the algorithm exploits the history of previous predicting performances and informs future strategies as will be discussed next. **Estimation of prediction uncertainty:** To design an online-algorithm that makes better decisions we need to determine the quality of the prediction. Knowing the prediction accuracy and robustness of a model can aid in defining how much we can trust our prediction model in future decision. In this approach, the accuracy of the prediction model is applied to choose the optimal hyper-parameters, $\lambda \in (0,1)$, which is a function of the scale-independent, Mean Absolute Percentage Error (MAPE), which is defined as [19]

$$\text{MAPE}\,(\%) = \frac{1}{H} \sum_{t=1}^{H} \left| \frac{y - \widehat{y}}{y} \right|, \tag{6}$$

where H is the history length pulled from a particular slice. λ is directly proportional to the input MAPE as shown in Fig. 4.

Algorithm Description: The input to the algorithm is the ground truth y, MAPE and the cost of the slice r. For each slice, the algorithm chooses the hyper-parameter λ according to MAPE (as in equation 6). In turn, λ and r determine the probability distribution p_i. The probability distribution defines the slot index at which the algorithm determines when the tenant should decide to buy to the end of the slice duration.

If the ML model predicts that $\widehat{y}_j \, u_j \geq r_j$ then the algorithm rents till time $v-1$ and buys at v according to the probability p_i for $k \leftarrow \lfloor r_j \, \lambda \rfloor$. If the ML model predicts that $\widehat{y}_j \, u_j < r_j$ then the algorithm rents till time $v-1$ and buys at v according to the probability p_i for $k \leftarrow \lceil r_j / \lambda \rceil$. The procedure is repeated for each slice, with the cost of the algorithm is determined as in line 14-18.

V. Results and analysis

In this section, we evaluate our methods by analyzing cost and utilization efficiency via simulations.

Algorithm 2: Probabilistic

Input: Let $\widehat{\mathbf{y}}$ be predictions of $\mathbf{y}$, $\mathbf{r}$ is the cost of buying the entire slice and $\mathbf{u}$ is the cost per slot.
Output: $\mathcal{A}$
for $j = 1, 2, \ldots, |\mathcal{S}|$ **do**
  while $T \neq 0$ **do**
    Calculate λ from equation (6)
    if $\widehat{y}_j \, u_j \geq r_j$ **then**
      $k \leftarrow \lfloor r_j \, \lambda \rfloor$.
    else
      $k \leftarrow \lceil r_j / \lambda \rceil$.
    for $i = 1, 2, \ldots, k$ **do**
      $p_i \leftarrow \left(\frac{r_j - 1}{r_j}\right)^{k-i} \times \frac{1}{r_j \left(1 - (1 - \frac{1}{r_j})^k\right)}$,
    Select: $v \in \{1, 2, \ldots, k\}$ according to probability p_i, where $\sum p_i = 1$.
    Define: $\beta = \left(u * \sum_{l=1}^{v-1} \mathbf{1} \cdot (d_{t_l} > 0)\right)$, where d_{t_l} is the real-time demand per slot and $\mathbf{1}$ is the indicator function.
    if $v \leq N$ **then**
      $a_j = \beta + r_j$
    else
      $a_j = \beta$
  $\mathcal{A} = \mathcal{A} + a_j$

A. Reducing allocation cost

One of the goals of dynamic RAN slicing is to reduce expenditures and to fulfill the increasing demand. We measure this performance metric by analyzing the cost of allocation strategy using the two approaches. To simplify the analysis, we assume the cost of buying an entire slice at any point during time window T is fixed, while the cost of one unit slot is 1 monetary unit. The demand is varied using uniformly distributed pseudo-random numbers, over 10^3 independent time windows and each time window consist of 200 slots. The predicted number of slots is simulated as $\widehat{y} = y + \delta$ where δ is generated from a normal distribution with zero mean and some standard deviation. Fig. 5 (left) depicts the results of our experiments for MAPE is 20%. The top figure represents the cumulative distribution function (CDF) of the aggregated cost using the two approaches. We observe that overall the probabilistic approach reduces the resource cost by approximately 12%. This is due to the improved allocation strategy, owing to an enhanced prediction.

The box plots show the cost per slice. Except for Slice 1, the results highlight the advantage of using the probabilistic approach over the baseline. As the sparsity in demand increases, the cost incurred using the probabilistic approach is lower than the baseline, as can be seen in Slice 2 and even more in Slice 3. The same behavior can be observed when MAPE is 60%, although with varied costs. See Fig. 5 (right).

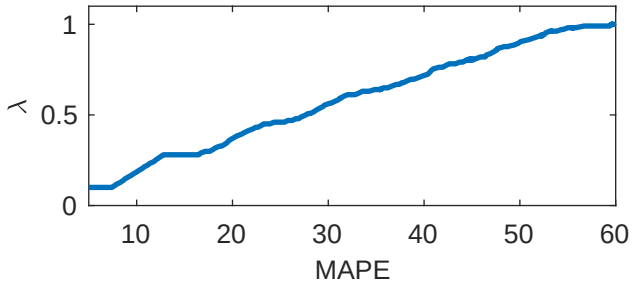

Fig. 4: MAPE vs λ.

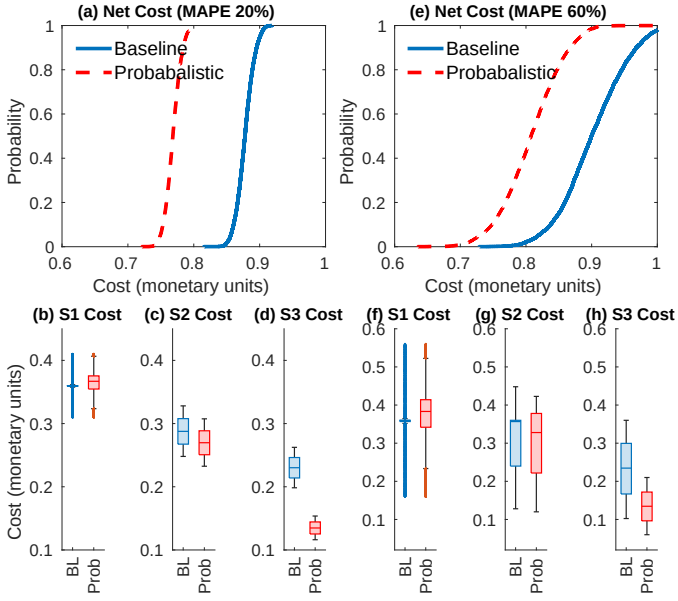

Fig. 5: The performance of baseline and probabilistic algorithm given (left) MAPE = 20% and (right) MAPE = 60%. Results are averaged over 10^3 trials.

B. Efficiency of Slice Allocation

Besides fulfilling the SLA requirements and reducing costs, the slicing allocation strategy must ensure maximum utilization of scarce radio resources. Therefore, it is necessary to evaluate the performance gains in terms of utilization. Slice allocation efficiency can be evaluated by the number of acquired slots, using either of the algorithms described above, against the number of utilized slots by the tenant (actual demand). In both algorithms, we use the same distribution of standard deviations away from the ground truth for a fair comparison. In this experiment, like the preceding one, we assume that there are three slices which can be created from a finite resource pool. We further assume that slices have an equal number of resource blocks. In each time slot, we observe the demand and the allocation of each algorithm across the three slices. We also vary MAPE to gain insights on its impact. Fig. 6, presents a CDF of the slice utilization under two different prediction uncertainties (Fig. 6 (left) MAPE = 20% and Fig. 6 (right) MAPE = 60%). It is evident from the two plots that the probabilistic approach can provide higher utilization and the difference between the two methods is more distinct when the MAPE is higher.

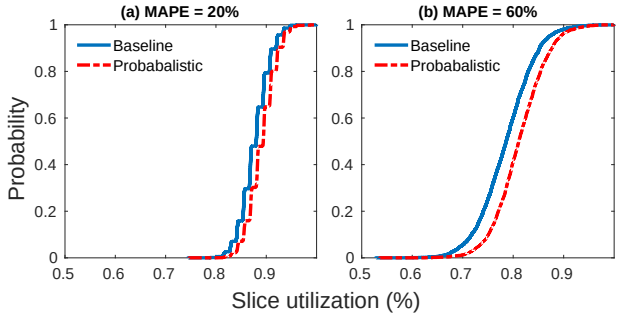

Fig. 6: Aggregated utilization of three slice for (left) MAPE = 20% and (right) MAPE = 60% over 5000 iterations. In each iteration we vary the demand y and the prediction $\widehat{y}$.

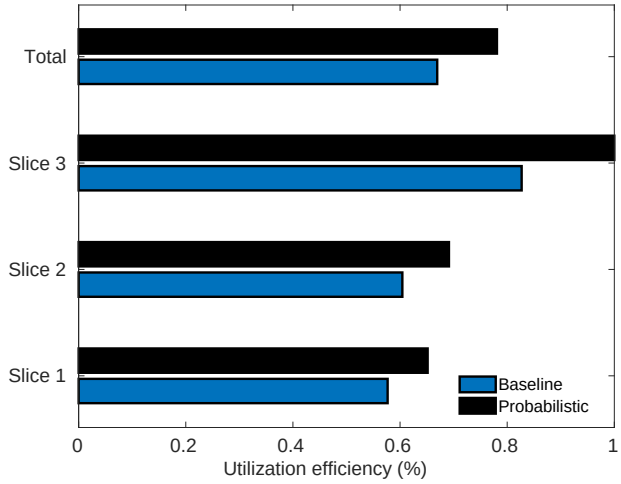

Fig. 7: Per slice utilization for MAPE = 60% (averaged over 5000 iterations). In each iteration we vary the demand y and the prediction $\widehat{y}$.

We also looked at the average individual utilization in each of the slices (See Fig. 7). Slice 3 is 100 percent utilized when using the probabilistic algorithm. This is due to the lower demand in Slice 3 in contrast to the demand from the other two slices, and the algorithm accurately requests slices based on real-time demand. In practice, we envisage that this will be below the 100% mark, since even real-time demand can exhibit inaccuracy, albeit with less severity. Head-to-head comparisons of the three slices show that the probabilistic approach gives higher utilization of resources than the baseline approach. This can be explained by the fact that the probabilistic approach is more resilient to error variance when it is prominent. Note that in this experiment we set MAPE = 60%.

VI. Conclusion

In this paper, we compared two online algorithms to configure RAN slicing to respond to real-time dynamic heterogeneous requirements. We showed that the probabilistic online algorithm can complement existing preliminary offline machine learning models, offering improved robustness in the presence of imperfect prediction. The analytical results show that for imperfect demand look-ahead, the probabilistic approach, whose performance is less sensitive to the prediction errors, can achieve lower costs to tenants and can provide higher resource utilization. Because of the increased importance of optimizing the costs in the RAN slicing, we believe many other deployments in various settings can benefit from our findings. In future work, we will further expand our models to include multiple tenants with varied demands who are sharing radio resources in order to analyze slice utilization, impact on SLA violations and fairness.

References

[1] CISCO, "Cisco annual internet report (white paper)," 2020.

[2] 3GPP, TS 23.501, v2.0.1 , "3GPP, TS 23.501 system architecture for the 5G system (Release 15)," 2017.

[3] G. H. Sim *et al.*, "An online context-aware machine learning algorithm for 5G mmWave vehicular communications," *IEEE/ACM Transactions on Networking*, vol. 26, no. 6, pp. 2487–2500, 2018.

[4] A. Kaloxylos, "A survey and an analysis of network slicing in 5G networks," *IEEE Communications Standards Magazine*, vol. 2, no. 1, pp. 60–65, 2018.

[5] 3GPP TS 38.300, v15.0.0, "NR and NG-RAN overall description; Stage 2 (Release 15)," 2017.

[6] Z. A. Qazi, M. Walls, A. Panda, V. Sekar, S. Ratnasamy, and S. Shenker, "A high performance packet core for next generation cellular networks," in *Proceedings of the Conference of the ACM Special Interest Group on Data Communication*, 2017, pp. 348–361.

[7] A. Ksentini and N. Nikaein, "Toward enforcing network slicing on RAN: Flexibility and resources abstraction," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 102–108, 2017.

[8] C. Marquez *et al.*, "How should I slice my network?: A multi-service empirical evaluation of resource sharing efficiency," in *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. ACM, 2018, pp. 191–206.

[9] X. Chen, Z. Han, H. Zhang, G. Xue, Y. Xiao, and M. Bennis, "Wireless resource scheduling in virtualized radio access networks using stochastic learning," *IEEE Transactions on Mobile Computing*, vol. 17, no. 4, pp. 961–974, 2017.

[10] X. Foukas, N. Nikaein, M. M. Kassem, M. K. Marina, and K. 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*, 2016, pp. 427–441.

[11] M. Yang, Y. Li, D. Jin, L. Su, S. Ma, and L. Zeng, "OpenRAN: a software-defined RAN architecture via virtualization," *ACM SIGCOMM computer communication review*, vol. 43, no. 4, pp. 549–550, 2013.

[12] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[13] C. Zhang and P. Patras, "Long-term mobile traffic forecasting using deep spatio-temporal neural networks," in *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, 2018, pp. 231–240.

[14] S. Isaacman, R. Becker, R. Cáceres, S. Kobourov, M. Martonosi, J. Rowland, and A. Varshavsky, "Identifying important places in people's lives from cellular network data," in *International Conference on Pervasive Computing*. Springer, 2011, pp. 133–151.

[15] A. Aijaz, "Hap-sliceR: A radio resource slicing framework for 5G networks with haptic communications," *IEEE Systems Journal*, vol. 12, no. 3, pp. 2285–2296, 2017.

[16] Y. Xu, F. Yin, W. Xu, J. Lin, and S. Cui, "Wireless traffic prediction with scalable Gaussian process: Framework, algorithms, and verification," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1291–1306, 2019.

[17] A. Khanafer, M. Kodialam, and K. P. Puttaswamy, "The constrained ski-rental problem and its application to online cloud cost optimization," in *2013 Proceedings IEEE INFOCOM*. IEEE, 2013, pp. 1492–1500.

[18] S. Gollapudi and D. Panigrahi, "Online algorithms for rent-or-buy with expert advice," in *International Conference on Machine Learning*, 2019, pp. 2319–2327.

[19] S. Kim and H. Kim, "A new metric of absolute percentage error for intermittent demand forecasts," *International Journal of Forecasting*, vol. 32, no. 3, pp. 669–679, 2016.