Resource Reservation in Sliced Networks: An Explainable Artificial Intelligence (XAI) Approach

Pieter Barnard, Irene Macaluso, Nicola Marchetti, Luiz A. DaSilva

***Abstract*—The growing complexity of wireless networks has sparked an upsurge in the use of artificial intelligence (AI) within the telecommunication industry in recent years. In network slicing, a key component of 5G that enables network operators to lease their resources to third-party tenants, AI models may be employed in complex tasks, such as short-term resource reservation (STRR). When AI is used to make complex resource management decisions with financial and service quality implications, it is important that these decisions be understood by a human-in-the-loop. In this paper, we apply state-of-the-art techniques from the field of Explainable AI (XAI) to the problem of STRR. Using real-world data to develop an AI model for STRR, we demonstrate how our XAI methodology can be used to explain the real-time decisions of the model, to reveal trends about the model's general behaviour, as well as aid in the diagnosis of potential faults during the model's development. In addition, we quantitatively validate the faithfulness of the explanations across an extensive range of XAI metrics to ensure they remain trustworthy and actionable.**

***Index Terms*—Explainable Artificial Intelligence (XAI), Network Resource Management (RM).**

I. Introduction

In 5G, *network slicing* allows mobile network operators (MNOs) to lease portions of their network resources to third party tenants, such as mobile virtual network operators (MVNOs), who seek to provide dedicated network services to their own end users (EUs) but lack the infrastructure to do so [1]. At the heart of its success, network slicing relies on the MNOs' ability to share their limited resources among multiple tenants in a flexible and efficient manner. While high flexibility can be achieved using technologies such as software-defined networking (SDN) and network function virtualisation (NFV) to create logical networks, or *slices*, that can be dynamically tailored towards the requirements of each tenant, doing so efficiently requires additional tools which can accurately model future resource demands when and where they are needed by each slice tenant [2].

In this paper, we examine the task of optimising the efficiency of a slice by means of performing short-term resource reservation (STRR) from the slice tenant's perspective. In contrast to our previous work [3], which looks at the accuracy trade-offs of various artificial intelligence (AI) models for STRR, this paper explores the *explainability* of AI models for STRR. In particular, our work is motivated by the fact that many AI models, such as deep neural networks (DNNs), are commonly perceived as "black-boxes" due to a lack of transparency in their behaviour [4]. In the context of STRR, this lack of transparency presents risks to both the reliable operation of the network, as well as potential revenue loss for operators should their STRR model, which leases resources autonomously in real time, fail and behave in an unexpected manner during its online operation. Recently, a suite of Explainable AI (XAI) techniques have emerged to aid the understanding of complex models by providing a human-in-the-loop with insights describing *why* and *how* the model arrives at its final decisions [5].

Although numerous studies have looked at the use of AI for STRR [2], [6]–[9], to the best of our knowledge, none have addressed the open problem of providing explainability to these solutions. Within the general slicing literature, only limited research pertains to the use of XAI; in [10], the authors outline the need for explainability in future 6G networks and provide a brief overview of how they envision XAI can be applied to handover and resource allocation problems. In [11], the authors compare the use of various XAI methods on a service-level agreement (SLA) violation model. Their results found that the SHapley Additive eXplanation (SHAP) method [12] provides the most consistent results when cross-referenced against a causal analysis tool. However, beyond a qualitative assessment, their analysis does not consider any quantitative metrics to assess the faithfulness of the explanations.

This paper is the first to apply an XAI methodology to network slicing problems, such as STRR, which quantitatively assesses the faithfulness of the generated explanations. In addition, we also demonstrate how explanations can be leveraged beyond just explaining the mere logic of the model, but also to aid in diagnosing potential faults that occur during the development of the model. In summary, we make the following key contributions in our work:

i) We outline a methodology based on state-of-the-art XAI techniques, such as Kernel SHAP [12] and global explanation methods [13], to explain the behaviour of AI models used in network slicing problems, such as STRR.
ii) We construct an STRR model using real-world data and demonstrate how explanations can be used to monitor the real-time decisions of the model, to reveal global trends about the model, as well aid in the diagnosis of potential faults during its development.
iii) We quantitatively validate the faithfulness of our explanations using an extensive range of XAI metrics [14].

Pieter Barnard (barnardp@tcd.ie), Irene Macaluso (macalusi@tcd.ie), Nicola Marchetti (nicola.marchetti@tcd.ie) is at CONNECT Research Centre, Trinity College Dublin, Ireland.

Luiz A. DaSilva (ldasilva@vt.edu) is at Commonwealth Cyber Initiative, Virginia Tech, USA

This publication has emanated from research conducted with the financial support of Science Foundation Ireland under Grants No. 18/CRT/6222 and 13/RC/2077_P2. We also acknowledge support from the Commonwealth Cyber Initiative (CCI).

The remainder of this paper is organised as follows: in Section II, we outline the details of our XAI methodology for STRR models. In Sections III and IV, we outline the system model for STRR in a sliced 5G network and discuss the details of our AI solution for STRR, respectively. In section V, we validate the performance of our STRR model, demonstrate examples of local and global explanations of our model, and finally, we evaluate the faithfulness of the explanations. In section VI we offer our conclusions and discuss possible avenues for future work.

II. XAI Methodology

As the basis of our methodology, we leverage the Kernel SHAP method from [12], which offers a strong theoretically-grounded framework for computing *local* explanations of a black-box model, i.e., explanations describing a model's behaviour around a single input of interest, and for which an open-source implementation is available for the Python language. In addition, we also consider techniques which provide *global* or general insights about a model [13], as well as techniques to evaluate the faithfulness of the explanations [14] in our methodology. Below, we briefly describe how each of these techniques pertain to our STRR model.

In Kernel SHAP, explanations take the form of *additive feature attribution*s (AFAs), which reflect the magnitude and direction by which each feature impacts a particular decision of the model. For example, if we consider a simplified STRR model which takes as input a tenant's past traffic demand and undelivered traffic, and outputs the amount of resources the tenant should reserve in the near-future, i.e., in units of Mb, then a local explanation may indicate, for instance, that knowledge of the past traffic demand has caused the model to increase its reservation by +12 Mb, while knowing the undelivered traffic has reduced it by -8 Mb.

In general, if we consider an STRR model which takes as input an N-dimensional vector of features, $\mathbf{x} = [x_1, ..., x_N]$, and produces a single output, $y = f(\mathbf{x})$, then a local explanation comprises of a set of attributions, or "SHAP" values, $\{\phi_1, ..., \phi_N\}$, where each x_i is associated with its own particular ϕ_i. Moreover, the SHAP value, ϕ_i, associated with feature x_i can be interpreted as the change in the expected output of the model when x_i is known to the model versus when x_i is unknown to the model. Finally, we note that the SHAP values derive their form from the Shapley value concept in game theory, and exhibits a number of desirable properties in contrast to most other XAI methods found in the literature [14]. For example, the "local accuracy" property ensures that the sum of all SHAP values for a specific instance adds up to the *difference* between the model's output for that instance and the expected output of the model, i.e.,

$$\sum_{i=1}^{N} \phi_i = y - E[f(\mathbf{x})]. \tag{1}$$

To generate global explanations of our model, we consider the use of SHAP dependence plots in our work. Essentially, a SHAP dependence plot can be used to understand the marginal effect a feature has on the output of the model, and is constructed for a particular feature by plotting its SHAP value (vertical axis) against the value that feature takes (horizontal axis), and repeating across multiple instances of the data. In the context of STRR, the resulting plots can help a human-in-the-loop to verify that the trained model correctly learns relationships experienced in the physical world, such as an STRR model that monotonically increases its reservation as the past traffic demand increases. In addition, global explanations can also aid in revealing potential bugs which may be harmful to the model's performance: in Section V, we demonstrate an example of how SHAP dependence plots allowed us to debug flaws in a preliminary, naive STRR model. We note that, while the use of XAI for model debugging has seen growing interest from within the general XAI community in recent years [15], its application in the telecommunications domain remains largely unexplored to date.

As the final step in our methodology, we validate the faithfulness of the explanations by considering an extensive range of XAI metrics. As outlined in [14], the majority of these metrics operate on a common basis, whereby the most important features are 'removed' from the model, i.e., by mean-masking features with the highest or lowest corresponding SHAP values, and observing the impact this has on the output of the model. Essentially, these metrics can be used to compare the relative performance of different XAI methods against one another. However, as we elaborate further in Section V, the majority of these metrics do not provide an absolute scale during their evaluation, which becomes essential whenever the faithfulness of the explanations are to be assessed. In our work, we compute this absolute scale by exhaustively permuting across all possible combinations of feature importance rankings, which form the basis of how these metrics operate.[1] The final result allows us to compare how well each explanation performs compared to the absolute score, and therefore, whether or not the explanation should be trusted and acted upon by a human-in-the-loop.

III. System Model

In this section, we outline the system model of a sliced network, which we subsequently use during the formulation of our AI solution for STRR. We begin by considering a sliced network comprising a single MNO serving the traffic demands of a set of tenants, K, across its network infrastructure. Assuming the MNO allocates its network resources within discrete scheduling windows, we outline four tasks which the MNO performs during a scheduling window at time t:

(a) *Schedules Future Resources:* The MNO receives reservation requests from each tenant, where $r_{t+i,k}$ denotes the amount of resources, expressed in units of Mb, which tenant $k \in K$ wishes to reserve during future time slot $t+i$. Provided the total sum of requested resources from all tenants for time $t+i$ does not exceed the MNO's maximum capacity, c, the MNO schedules the requested resources of each tenant for future use. If the sum of requested resources exceeds the maximum capacity, the MNO distributes its resources proportionally amongst

[1] We note that this process requires complexity of the order $\mathcal{O}(NN!)$.

the tenants such that the actual amount of resources scheduled for tenant k at future time $t+i$, is given by:

$$a_{t+i,k} = \begin{cases} r_{t+i,k}, & \sum_{k \in K} r_{t+i,k} \leq c \\ \dfrac{c \cdot r_{t+i,k}}{\sum_{k \in K} r_{t+i,k}}, & \text{else} \end{cases} \tag{2}$$

(b) *Distributes Best-Effort Traffic:* The MNO distributes any spare capacity it has among the under-provisioned tenants in a best-effort manner. Using $v_t = c - \sum_{k \in K} a_{t,k}$ to denote the MNO's spare capacity at time t, $s_{t,k}$ to represent the actual traffic demands of tenant k at time t, and $h_{t,k} = max(s_{t,k} - a_{t,k}, 0)$ to signify the amount by which tenant k is under-provisioned at time t, the best-effort traffic assigned to tenant k is given by:

$$b_{t,k} = \begin{cases} 0, & h_{t,k} = 0 \\ min\left(h_{t,k}, \dfrac{v_t}{|K|}\right), & \text{else} \end{cases} \tag{3}$$

(c) *Charges Each Tenant:* Tenants are charged according to their future reserved resources and those which have been assigned to them as best-effort. Although specific pricing strategies are beyond the scope of this work, we assume that the unit cost of reserved resources is much higher than for those allocated in a best-effort manner. Thus, in our STRR model (see Section IV), a tenant aims to minimise the price it pays for resources, while maximising the quality of service (QoS) to its EUs, by reserving only the amount of resources that are in excess of what it expects to be available in a best-effort manner.

(d) *Provides Feedback:* At the end of the scheduling window, the MNO provides feedback information to each tenant regarding the network statistics of their own slice. This includes their total traffic demand $s_{t,k}$, best-effort traffic $b_{t,k}$, undelivered traffic $u_{t,k}$, delivered traffic $d_{t,k}$ and the number EUs that are active in their slice $z_{t,k}$. Using equations (2) and (3), the delivered and undelivered traffic of tenant k can be derived as:

$$d_{t,k} = min(a_{t,k} + b_{t,k}, s_{t,k}) \tag{4}$$

and

$$u_{t,k} = s_{t,k} - d_{t,k}. \tag{5}$$

IV. A Deep Learning Solution

In this section, we describe the details of our STRR model. We begin by formulating the reservation task, in which a slice tenant, k, reserves resources from the MNO to meet the future demands of its EUs. Specifically, the tenant wishes to minimise its expenditure for resources (i.e., by avoiding over-reservations), while maximising the QoS to its EUs (i.e., by avoiding under-reservations). We cast this problem into a supervised learning task, where the objective is to achieve a good trade-off between expenditure for resources and QoS. Using the aggregated traffic traces of each tenant and system model from Section III, we construct the ground-truth labels of our model by simulating the behaviour of an *oracle*, which has full knowledge of the network's parameters and future states, and makes informed reservations for our tenant k, given by:

$$r^*_{t,k} = \begin{cases} 0, & \dfrac{c - l_t}{|K|} \geq s_{t,k} \\ min\left(s_{t,k}, \dfrac{|K| s_{t,k} + l_t - c}{|K| - 1}\right), & \text{else} \end{cases} \tag{6}$$

where for brevity, we use l_t to denote the total sum of reservations from all *other tenants* excluding our own, i.e.,

$$l_t = \sum_{i \in K \setminus \{k\}} r_{t,i}. \tag{7}$$

Intuitively, this policy urges our tenant not to make reservations when best-effort resources are large enough to cover its future traffic demand. When the availability of best-effort resources is limited, i.e., due to heavy competition in the network, the model will at most reserve the tenant's own future demand, or else the minimum portion of it that cannot be covered by the resources available on a best-effort basis.[2] The outcome of this simulation produces the ground-truth labels and corresponding signals $s_{t,k}$, $u_{t,k}$, $b_{t,k}$, $d_{t,k}$, and $z_{t,k}$, which our tenant receives when making reservations under this policy. We note that for the purpose of this simulation, we assume that the other remaining tenants are perfect predictors of their own traffic, and thus make reservations according to $r_{t,i} = s_{t,i}, i \in K \setminus \{k\}$. We also set the maximum capacity of the network equal to the 80th percentile of the total traffic occurring during the training data, i.e., $c = P_{80}(\sum_{i \in K} s_{t,i})$.

For the design of our model, we implement a fully-connected DNN with 3 hidden layers between the input and output layer. As inputs to the model, we consider six sources of information available to our tenant at time t, including its most recent traffic demand, best-effort traffic, undelivered traffic, number of EUs, as well as two time-based features capturing seasonal trends at a daily and weekly level by considering the number of minutes passed since the start of each day (i.e. 00:00) and start of each week (i.e., Monday), respectively.[3] We also standardise each feature to have zero mean and unit variance before being passed as inputs to the model. Finally, the output of the model consists of a single scalar representing the reservation for the next scheduling window at time $t+1$.

To train our model, we use the Keras and TensorFlow libraries for Python. We optimise the parameters of the model using Stochastic Gradient Descent (SGD) with a Huber loss function. This loss function is characterised by a 'delta' parameter which specifies a threshold at which the loss changes from a quadratic to a linear scale; as we find that the traffic data exhibits rare peaks that harm the model's convergence, the Huber loss presents a better alternative to the conventional L2 loss, which is known to be sensitive to outliers in the data. To determine a suitable value for this parameter, along with other hyper-parameters of the model, such as the number of

[2]Although other reservation policies are also feasible in this context, such as [3], the choice of policy is arbitrary and does not affect the primary concern of our work, which is to provide explainability to the underlying model.

[3]We do not include the tenant's delivered traffic as a feature, as this becomes redundant to the model once the total and undelivered amounts are given.

neurons per layer, activation function (sigmoid, ReLU or tanh), and dropout rate for each layer, we perform a grid search of the model space using the Bayesian optimiser from the Keras Tuner library. Our final model consists of 5 layers, including the input layer, 3 hidden layers of sizes 16, 256 and 112 neurons, respectively, and an output layer. For each of the hidden layers, we use a ReLU activation function, as well as apply drop out of strength 0.1, 0.5 and 0.1, respectively. We train our final model for 1000 epochs using an early stop criteria of 50 epochs on the validation loss, and use mini-batches of size of 288 samples in order to improve the convergence rate during training.

We use a publicly available dataset [16] of cellular traffic data recorded from over 5,000 mobile phone users during 2014. Each record contains application-layer data about the EUs' traffic, which can be mapped directly to their own service slice. To use this dataset in our work, we first aggregate each user's traffic trace into 5 minute intervals (the length of a scheduling window in our model) and combine the uplink and downlink traces of each application into a single trace. The resulting dataset gives us the traffic demands (used to compute the ground-truth labels in Eq. (6)) and number of EUs that are active in each slice during each scheduling period. Finally, we use the 'Chrome' trace from this dataset as the tenant used by our model, and split the dataset into three sections, using approximately eight weeks of data for training (September 1st up to October 24th), one week for validation (October 25th up to October 31st), and one month for testing (November 1st up to November 30th).

V. Results

In this section, we evaluate the performance of our STRR model, and demonstrate the results of our XAI methodology applied to the model.

A. Performance of STRR model

We evaluate the performance of our model using a series of key performance indicator (KPI) metrics, such as the root mean-square error (RMSE), and over/under reservation, as defined by equations (8) and (9), respectively, where $t \in T$ refers to samples taken from the test set and $\mathbb{1}\{\cdot\}$ denotes the indicator function. For comparison, we also show the performance of a naive solution, which always reserves the most recent traffic demand for the next scheduling window. As shown in Table I, our model is able to outperform the naive solution in terms of RMSE and over-reservation but slightly under performs in terms of under-reservation.[4]

$$Over = \frac{\sum_{t=1}^{|T|} (\hat{r}_{t,k} - r^*_{t,k}) \mathbb{1}\{\hat{r}_{t,k} > r^*_{t,k}\}}{\sum_{t=1}^{|T|} \mathbb{1}\{\hat{r}_{t,k} > r^*_{t,k}\}} \quad (8)$$

$$Under = \frac{\sum_{t=1}^{|T|} (\hat{r}_{t,k} - r^*_{t,k}) \mathbb{1}\{\hat{r}_{t,k} < r^*_{t,k}\}}{\sum_{t=1}^{|T|} \mathbb{1}\{\hat{r}_{t,k} < r^*_{t,k}\}} \quad (9)$$

[4]It is important to emphasise here that the primary purpose of our work concerns the explainability of our model, as opposed to its shear performance. We therefore leave further efforts to improve its performance for future work.

TABLE I: Model Performance

Metric	Model	Naive
RMSE (Mb)	**16.3**	19.1
Over-Reservation (Mb)	**9.2**	11.5
Under-Reservation (Mb)	-13.1	**-11.8**

B. Local Explanations

Fig. 1 illustrates an example of a local explanation of our STRR model. In this example, each bar corresponds to the SHAP value of one of the model's inputs, where red bars indicate features that raise the model's output above its expected value of $\approx$ 17.1 Mb, while blue bars indicate features that lower it. Specifically, the explanation for this sample determines that the time of day/week, number of EUs, and past traffic demand, drive the model's output above its expected value, with past traffic being responsible for the highest increase of $\approx$ 5.2 Mb. On the other hand, best-effort and undelivered traffic lower the output of the model, with best-effort lowering the output the most by $\approx$ 5.9 Mb.

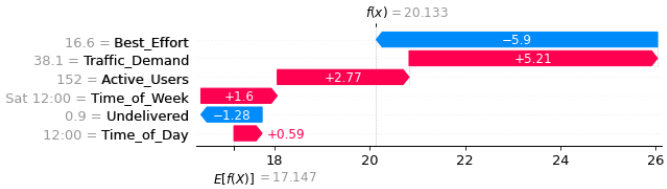


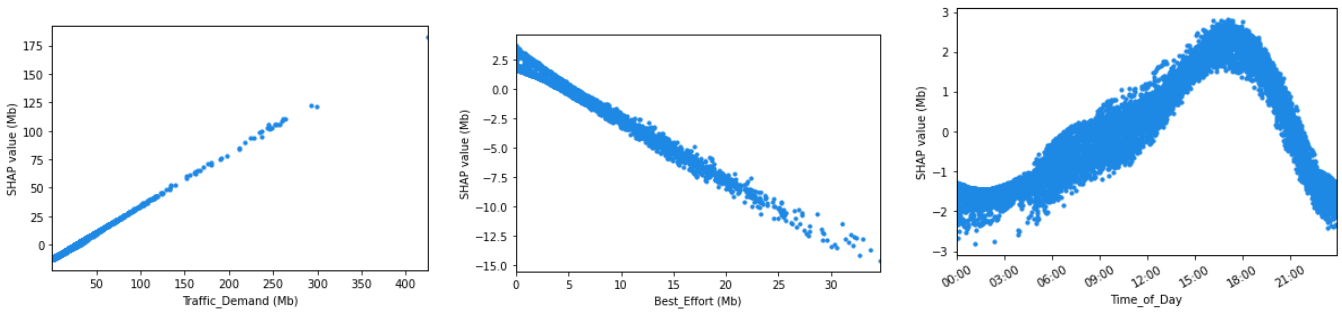
Fig. 1: Example local explanation using Kernel SHAP.

Importantly, in the context of STRR, local explanations can be used to *monitor* and *verify* the real-time decisions of the STRR model. As an example of how this can be achieved in the case of our own STRR model, we start by considering the tenant's traffic demand of 38.1 Mb (as shown in Fig. 1), which in this case can be considered relatively high compared to the average traffic demand of $\approx$ 26.7 Mb. In this sense, as the explanation indicates that knowledge of the traffic demand has led to an increase of the model's reservation, this suggests that the model has interpreted high traffic demand to be associated with reduced network availability, which the model appropriately compensates for by increasing its reservation so as to minimise any service loss. Similarly, by extending this analysis towards the remaining features of the model, it can be verified that the model appropriately increases its reservation based on the current time of day/week and relatively high number of EUs, while the relatively high best-effort and low undelivered traffic reduce the reservation amount accordingly.[5]

C. Global Explanations.

Due to limited space, we illustrate SHAP dependence plots only for the tenant's past traffic demand, best-effort and time of day features, as shown in Fig. 2. By examining these

[5]We note that the analysis shown here applies only to our particular example. In general, the manner in which one might verify the decisions of their model will depend on the nature of the problem as well as the domain knowledge of the human-in-the-loop performing the analysis.



(a) (b) (c)

Fig. 2: SHAP dependence plots of our STRR model.

plots, a number of important insights can be gained about the model's general behaviour. Specifically, we see in Fig. 2a that the model's reservations tend to increase monotonically as the past-traffic demand increases, while in Fig. 2b, we see that the model monotonically decreases its reservations as the best-effort increases. Finally, we see in Fig. 2c that the model exhibits a cyclic relationship with the time of day feature, whereby its reservations are reduced during the early/late hours of the day, and peak towards the afternoon/evening.

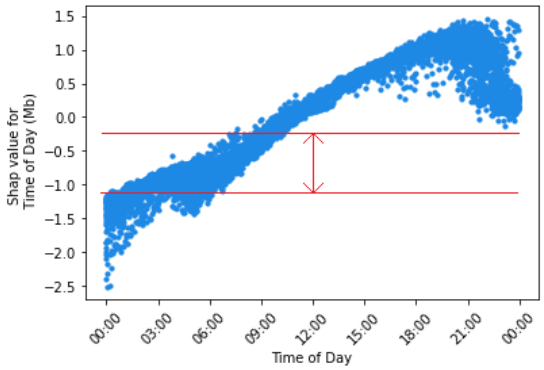


Fig. 3: SHAP dependence plot revealing a concerning disparity associated with the start and end of each day for the time of day feature.

Importantly, global explanations can also help identify and debug potential flaws in a model before its final deployment. As an example of how global insights were used for model debugging during the development of our own STRR model, we consider the SHAP dependence plot shown in Fig. 3, which corresponds to the time of day feature taken during the early design stages of our model. In contrast to the behaviour of this feature within our final model, as depicted in Fig. 2c, Fig. 3 reveals a significant disparity between the start and end of each day, which we consider as concerning given that these two time stamps occur consecutively after one another. [6] We note that this problem was overcome in our final model by applying the 'trigonometric' transformation method described in [17] to both the time of day and time of week features, as given by expressions (10) and (11), respectively.

[6]Although omitted here due to space constraints, similar behaviour was also observed in the time of week feature.

$$X_{Day} \longrightarrow sin\left(\frac{2\pi X_{Day}}{1440}\right), cos\left(\frac{2\pi X_{Day}}{1440}\right), \tag{10}$$

$$X_{Week} \longrightarrow sin\left(\frac{2\pi X_{Week}}{10080}\right), cos\left(\frac{2\pi X_{Week}}{10080}\right). \tag{11}$$

D. XAI Evaluation

We validate the faithfulness of our explanations using the following metrics from [13]:

(a) *Local Accuracy:* This metric measures the ability of an explanation method to produce importance scores ϕ_i that sum up to the original output of the model $f(\mathbf{x})$ for the sample being explained. Similar to [14], we evaluate the local accuracy of our explanations by computing the normalized standard deviation of the difference between the feature importance scores and model output across 100 random samples from the test set using equation (12):

$$\sigma = \frac{\sqrt{E_x[(f(\mathbf{x}) - (\sum_{i=1}^{N} \phi_i + E[f(\mathbf{x})]))^2]}}{\sqrt{E_x[f(\mathbf{x})^2]}}. \tag{12}$$

To determine how well an explanation method performs on this metric, [14] proposes the use of nine cutoff regions within the range between 1.00 (highly accurate) and 0.10 (poorly accurate).

(b) *Mask-Based Metrics:* This refers to a suite of metrics that collectively assesses the ability of an explanation to identify features that increase/decrease the output of the model the most, and those with the largest impact to the model's accuracy. For example, to compute the Keep Positive metric, which assesses how well an explanation identifies features that increase the output of the model the most, we first: i) 'remove' all inputs to the model by masking each feature with its mean value, and observe the output of the model; ii) for increasing fractions of features, we unmask the features with the most positive SHAP values, and observe the output of the model at each fraction; and iii) plot the fraction of features kept (x-axis) against the model output (y-axis). In this sense, a good explanation should produce a curve which shows a large increase at lower fractions, while levelling off at

higher fractions to produce a large area under the curve (AUC) (see Fig. 4).[7]

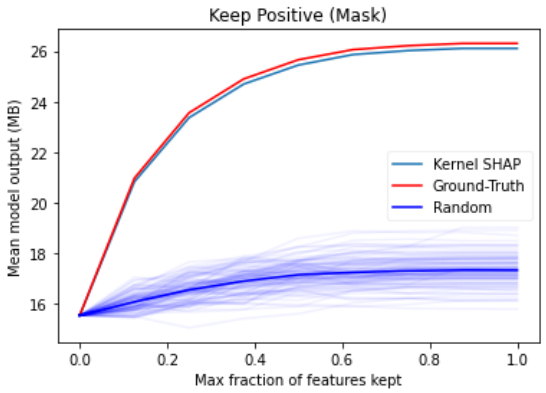


Fig. 4: Performance curve of our XAI methodology on the Keep Positive metric.

Similar to [14], we compute each of the mask-based metrics across 100 random samples from the test set. To evaluate the faithfulness of the explanations, we calculate the average AUC of each metric and compare it to its average absolute AUC, where, for a single sample, we obtain the absolute AUC by permuting across all combinations of relative feature importance, and taking the permutation which yields the lowest/highest AUC, as appropriate to the particular metric being assessed. For further comparison, we also consider the performance of a 'random-guessing' approach, whereby importance is assigned randomly to each feature, and take the average AUC following 100 trials per sample.

In Table II, we summarise the performance of our methodology under each of the these metrics:

TABLE II: Summary of Evaluation Performance

Evaluation Metric	Model	Absolute	Random
Local Accuracy (δ)	1.0	1.0	-
Keep Positive*	68.77	70.2	10.66
Remove Negative*	62.04	63.18	3.28
Remove Absolute*	82.51	92.12	48.57
Remove Positive*	16.98	16.05	71.26
Keep Negative*	7.49	6.34	64.42
Keep Absolute*	26.32	16.45	62.14

* Measured in units of area.

The results in Table II indicate that the explanations used in our work achieve near optimal performance in almost all considered cases, with the worst achieving metric (Keep Absolute) still yielding significant improvement over the random case. In practice, this would signify that the explanations can be trusted and acted upon with high confidence.

[7]In the case of the Remove Positive metric, which starts by keeping all features unmasked and removing increasing fractions of features with the most positive SHAP values, the resulting curve should decrease the greatest at lower fractions of features removed, thus, yielding a small AUC.

VI. Conclusion

In this work, we apply an XAI methodology based on state-of-the-art techniques to demystify the behaviour of complex AI models used in STRR within sliced networks. Using real-world traffic data, we develop our own STRR model, and demonstrate how explanations can be used to monitor and verify the real-time decisions of the STRR model, to reveal important trends about the model's general behaviour, as well as diagnosing potentially harmful behaviours before the model is deployed in a real-world environment.

In future work, we hope to investigate the potential use of XAI in other telecommunication domains, such as security, anomaly detection, resource allocation etc., as well as any additional benefits that XAI can offer within these settings, such as detecting distributional shifts in the input data [14] or performing root-cause analysis [11].

References

[1] X. Foukas *et al.*, "Network slicing in 5G: Survey and challenges," *IEEE Commun. Mag.*, vol. 55, no. 5, pp. 94–100, 2017.

[2] D. Bega *et al.*, "Deepcog: Optimizing resource provisioning in network slicing with ai-based capacity forecasting," *IEEE Journ. Sel. Areas Commun.*, vol. 38, no. 2, pp. 361–376, 2019.

[3] J.-B. Monteil *et al.*, "Resource reservation within sliced 5g networks: A cost-reduction strategy for service providers," in *2020 IEEE Int. Conf. Commun. Work. (ICC Workshops)*. IEEE, 2020, pp. 1–6.

[4] F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," *arXiv preprint arXiv:1702.08608*, 2017.

[5] A. Adadi and M. Berrada, "Peeking inside the black-box: A survey on explainable artificial intelligence (xai)," *IEEE Access*, vol. 6, pp. 52 138–52 160, 2018.

[6] X. Shen *et al.*, "Ai-assisted network-slicing based next-generation wireless networks," *IEEE Open Journ. Vehic. Tech.*, vol. 1, pp. 45–66, 2020.

[7] F. Malandrino and C.-F. Chiasserini, "5G traffic forecasting: If verticals and mobile operators cooperate," in *2019 15th Ann. Conf. Wire. On-dem. Net. Sys. Serv. (WONS)*. IEEE, 2019, pp. 79–82.

[8] Q. Guo *et al.*, "Proactive dynamic network slicing with deep learning based short-term traffic prediction for 5g transport network," in *2019 Opt. Fibe. Commun. Conf. Exhib. (OFC)*. IEEE, 2019, pp. 1–3.

[9] G. Sun *et al.*, "Dynamic reservation and deep reinforcement learning based autonomous resource slicing for virtualized radio access networks," *IEEE Access*, vol. 7, pp. 45 758–45 772, 2019.

[10] C. Li *et al.*, "Trustworthy deep learning in 6g-enabled mass autonomy: From concept to quality-of-trust key performance indicators," *IEEE Vehic. Tech. Mag.*, vol. 15, no. 4, pp. 112–121, 2020.

[11] A. Terra *et al.*, "Explainability methods for identifying root-cause of sla violation prediction in 5G network," in *IEEE Global Commun. Conf. GLOBECOM*. IEEE, 2020, pp. 1–7.

[12] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Adv. Neur. Infor. Proc. Sys.*, 2017, pp. 4765–4774.

[13] S. M. Lundberg *et al.*, "Consistent individualized feature attribution for tree ensembles," *arXiv preprint arXiv:1802.03888*, 2018.

[14] S. M. Lundberg, G. Erion *et al.*, "From local explanations to global understanding with explainable ai for trees," *Nature machine intelligence*, vol. 2, no. 1, pp. 2522–5839, 2020.

[15] J. Adebayo *et al.*, "Debugging tests for model explanations," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 700–712.

[16] F. A. Silva, A. C. Domingues *et al.*, "Discovering mobile application usage patterns from a large-scale dataset," *ACM Trans. Knowl. Discov. Data (TKDD)*, vol. 12, no. 5, pp. 1–36, 2018.

[17] A. Adams and P. Vamplew, "Encoding and decoding cyclic data," *The South Pacific Journal of Natural Science*, vol. 16, 1998.