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

Pieter Barnard ¹, Irene Macaluso ², Nicola Marchetti ², and Luiz Pereira da Silva ²

¹Trinity College Dublin
²Affiliation not available

November 1, 2023

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.
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 autonomously leases resources in real time, fail and behave in an unexpected manner during its 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 exXplanation (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 a 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

In this section, we describe our XAI methodology, which we later use to explain the behaviour of our STRR model. As the basis of this methodology, we make use of the Kernel SHAP method [12]¹ to compute local explanations of our model, which are those that describe the model’s behaviour around a single input of interest. In addition, we also consider techniques that provide global or general insights into our model [13], as well techniques to evaluate the faithfulness of the explanations [14]. Below, we briefly describe how each of these techniques pertain to our STRR model.

Under the Kernel SHAP framework, local explanations are produced in the form of additive feature attributions (AFAs), which reflect how much each input feature impacts a particular decision of the model. For example, if we consider an STRR model, which takes as input two features corresponding to a tenant’s past traffic demand and undelivered traffic, while the output corresponds to the amount of resources the tenant should reserve in the near-future, then a local explanation may, for instance, indicate that the tenant’s past traffic demand contributes 6 Mb towards a final request of 20 Mb (i.e., a positive effect), while the undelivered traffic reduces the reservation amount by 8 Mb (i.e., a negative effect).

In general, if we consider an STRR model which maps an N-dimensional vector of input features, \( \mathbf{x} = [x_1, \ldots, x_N] \), to a single output, \( y = f(\mathbf{x}) \), a local explanation comprises of a set of attributions, or "SHAP" values, \( \{\phi_1, \ldots, \phi_N\} \), where each \( x_i \) is associated with its own particular \( \phi_i \). Moreover, the SHAP value, \( \phi_i \), 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. In addition, we note that the SHAP values derive their form from the Shapley value concept in game theory, and exhibit a number of desirable properties, including that of local accuracy, in which the sum of all SHAP values for a particular explanation adds up to approximate the difference between the model’s output and the expected output of the model, i.e.,

\[
\sum_{i=1}^{N} \phi_i \approx y - E[f(\mathbf{x})].
\] (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 individually for each feature by plotting the SHAP value of the feature (vertical axis) against the value the feature takes (horizontal axis), for 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 a STRR model which monotonically increases its reservation as the past traffic demand increases. In addition, global explanations can also aid in revealing potential bugs which are 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 the general XAI community [15] in recent years, 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 by 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 is 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 these metrics.² 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 with confidence 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 \( K \) tenants 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

¹We note that an open-source implementation of Kernel SHAP is available for the Python programming language.

²We note that this process requires complexity of the order \( O(NN!) \).
We begin by formulating the reservation task, in which a model from Section III, we construct the ground-truth labels of using the aggregated traffic traces of each tenant and system by avoiding under-reservations). We cast this problem into a supervised learning task, where the objective is to achieve a future demands of its EUs. Specifically, the tenant wishes aims to minimise the price it pays for resources, while maximising quality of service (QoS) by reserving only the amount of resources that are in excess of what it expects to be available in a best-effort manner. Thus, in our STRR model (see Section IV) a tenant makes informed reservations for tenant \( k \), given by:

\[
r^*_t, k = \begin{cases} 0, & h_{t, k} = 0 \\ \min \left( h_{t, k}, v_t \left| \frac{c}{|K|} \right. \right), & \text{else} \end{cases} \
\]

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

\[
l_t = \sum_{i \in K \setminus \{k\}} r_{t, i}.
\]

Intuitively, this policy urges a 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, 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.\(^3\)

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 \( r_{t, i} = s_{t, i}, i \in K \setminus \{k\} \), and also set the maximum capacity of the network to be 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, unreserved 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.\(^4\)

We trained our initial model using these six features as raw inputs to the model. However, as we elaborate on further in Section V-B, insights from our XAI methodology allowed us to detect harmful behaviour associated with the time of day/week features. Subsequently, our final model includes eight inputs, where the time of day/week features have each been mapped to two new features using expressions (8) and (9), respectively, based on the ‘trigonometric’ feature engineering method described in [16]. 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 occurring at time \( t + 1 \).

\(^3\)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.

\(^4\)We do not include the tenant’s delivered traffic as a feature, as this becomes redundant to the model once the total and unreserved amounts are given.
To train our model, we use the Keras and TensorFlow libraries for Python. To optimise the parameters of the model, we use 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 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 a single input layer, followed by 3 hidden dense layers of sizes 16, 256 and 112 neurons, each with a ReLU activation function and drop out rates of 0.1, 0.5 and 0.1, respectively, followed by the output layer. We train our final model for 1000 epochs using an early stop criteria of 50 epochs on the validation loss, and use mini-batching of size of 288 samples in order to improve the convergence time during training.

We use a publicly available dataset [17] 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, 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.

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 (10) and (11), respectively, where \( t \in T \) refers to samples taken from the test set and \( 1\{\cdot\} \) 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.

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

\[
\begin{align*}
O_{\text{ver}} &= \frac{1}{T} \sum_{t=1}^{T} (\hat{r}_{t,k} - r_{t,k}) \{\hat{r}_{t,k} > r_{t,k}^*\} - \frac{1}{T} \sum_{t=1}^{T} \{\hat{r}_{t,k} > r_{t,k}^*\} \\
U_{\text{nder}} &= \frac{1}{T} \sum_{t=1}^{T} (\hat{r}_{t,k} - r_{t,k}) \{\hat{r}_{t,k} < r_{t,k}^*\} - \frac{1}{T} \sum_{t=1}^{T} \{\hat{r}_{t,k} < r_{t,k}^*\}
\end{align*}
\]

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\(^6\), 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

\(^6\)We emphasise that the primary purpose of our work concerns the explainability of the model, as opposed to its performance, and we leave any additional effort to improve this model for future work.

\(^7\)Here, we have re-combined the SHAP values of the cosine/sine components associated with the time of day/week features into a single value.
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 reduces the reservation amount accordingly.7

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 plots, a number of important insights can be gained about the model’s general behaviour. Specifically, in Fig. 2a, we see 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, in Fig. 2c, we see that the model’s behaviour 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.

Moreover, to understand how global insights helped improve the reliability of our own model, we consider a SHAP dependence plot of the time of day feature constructed during the early stages of the model’s development, as shown in Fig. 3. In contrast to the final model, seen in Fig. 2c, Fig. 3 reveals a concerning discontinuity between the start and end of each day.8 Only after implementing the ‘trigonometric’ method (see Section IV), can we verify in Fig. 2c that the resulting model exhibits the expected behaviour, with SHAP values at the beginning and end of the day matching.

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 \( \phi_i \) that sum up to the original output of the model \( f(x) \) 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):

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

7We 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.

8Although omitted here due to space constraints, similar behaviour was also observed in the time of week feature.

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, we first: i) ‘remove’ all inputs to the model by mean masking, 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 the largest increase at lower fractions, while levelling off at higher fractions to produce a large area under the curve (AUC) (see Fig. 4).

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 (\( \delta \)) | 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
Fig. 2: SHAP dependence plots of our STRR model.

Fig. 3: SHAP dependence plot of the time of day feature showing large discontinuity between start and end of day.

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

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 wire-less 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. Fibre. 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. Neural 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] A. Adams and P. Vamplew, “Encoding and decoding cyclic data,” The South Pacific Journal of Natural Science, vol. 16, 1998.
[17] 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.