Explainable Artificial Intelligence for Energy-Efficient Radio Resource Management

Alexandru-Daniel Marcu*†, S. Krishna Gowtam Peesapati†, Jessica Moysen Cortes†, Sahar Imtiaz*, James Gross*
*KTH Royal Institute of Technology, Stockholm, Sweden
†Huawei Technologies Sweden AB, Stockholm, Sweden
Email: {admarcu, sahari, jamesgr}@kth.se, {krishna.peesapati, jessica.moysen.cortes}@huawei.com

***Abstract*—As wireless systems evolve, the problems of radio resource management (RRM) become harder to solve. Once the additional constraint of energy-efficient utilization of resources is factored in, these problems become even more challenging. Thus, experts started developing solutions based on complex artificial intelligence (AI) models that, unfortunately, suffer from a performance-explainability trade-off. In this work, we propose an explainable AI (XAI) methodology for addressing this trade-off. Our methodology can be used to generate feature importance explanations of AI models through three XAI methods: (i) Kernel SHapley Additive exPlanations (SHAP), (ii) Counterfactual Explanations for Robustness, Transparency, Interpretability, and Fairness of Artificial Intelligence models (CERTIFAI), and (iii) Anchors. For Anchors, we formulate a new feature importance score based on the feature's presence within the rules built by the method. We then use the generated explanations to improve the understanding of the model and reduce its complexity through a feature selection process. By applying our methodology to a reinforcement learning (RL) agent designed for energy-efficient RRM, we were able to reduce its complexity by approximately $27\% - 62\%$ according to various metrics, without losing performance. Additionally, we show the possibility to replace the AI-based inference process with an Anchors-based inference process with similar performance and higher interpretability for humans.**

***Index Terms*—5G Networks, Energy Efficiency, Radio Resource Management, Reinforcement Learning, Explainable AI**

I. Introduction

Driven by increasing quality of service and quality of experience requirements, wireless systems have been continuously evolving. One of the most representative examples of this trend comes from mobile networks. The current fifth-generation (5G) networks enable higher data rates and connectivity density, lower latency, greater coverage and mobility than previous generations. To meet these requirements, an increased utilization of radio resources is needed (i.e., larger antenna arrays, more operating bands, wider bandwidths), which can result in an increased network energy consumption (EC) [1]. Therefore, it is important to manage these resources and adapt their utilization to the traffic demands to ensure an energy-efficient network [2], [3]. However, the problem of RRM becomes more challenging as the wireless networks evolve and the number of radio resources that need to be managed grows. Thus, experts started using advanced AI methods such as deep learning (DL) and RL to solve this problem [4].

Even though these AI methods can yield a good solution for RRM problems, they suffer from a performance-explainability trade-off [5]. This is also known as the complexity-interpretability trade-off of AI methods [6]. In other words, the more complex an AI model is, the harder it is for humans to understand its behavior in different scenarios, which can lead to a lack of trust in the AI model.

To mitigate the effects of this trade-off, one can resort to XAI. This field provides various methods for deriving explanations of the reasoning and results of AI models, making them more understandable and interpretable for humans. Some common types of explanations that XAI methods can generate include text, visual, and feature importance explanations. Moreover, these XAI methods can be classified using different criteria such as (a) the scope of the generated explanation (local and global methods), (b) the time of information extraction (intrinsic and post-hoc methods), and (c) the specificity of the method (model-specific and model-agnostic methods) [7].

In this work, we focus on three model-agnostic post-hoc XAI methods, namely Kernel SHAP, CERTIFAI, and Anchors, to generate different types of explanations (rules, counterfactuals, and feature importance explanations) of an RL agent for energy-efficient RRM. While the RL agent itself is used to solve the RRM problem, we aim to use these XAI methods, explanations, and in particular feature importance to:

- Improve the understanding of the RL agent by understanding its inferences;
- Simplify the RL agent through feature selection based on feature importance, without losing performance[1].

The main contributions of this paper are:

- We use a diverse set of XAI methods to determine and analyze the importance of different input features of an RL agent for energy-efficient RRM;
- We propose an approach to generate feature importance explanations using Anchors and CERTIFAI, and compare the three methods in terms of computation time and memory requirements needed to generate the explanations;
- We present an analysis of the performance of rule-based inference, and compare it to an RL-based inference process.

The rest of this paper is organized as follows. In section II, we discuss related work. The XAI methods are explained in section III. In section IV, we describe the RL agent given as

[1]Reducing the complexity of AI models lowers the cognitive load on the engineers using them and shortens the model's training time.

a use case. Our results are detailed in section V. We conclude and propose possible future work in section VI.

II. Related work

To the best of our knowledge, no papers have focused on generating feature importance explanations of AI models for RRM, or using these explanations to improve the understanding or reduce the complexity (e.g., through feature selection) of such models. Nevertheless, there are works that use feature importance explanations for similar purposes in other networking areas. In [8], Kernel SHAP is used to derive feature importance explanations of an AI model for short-term network resource reservation. It is shown that these explanations can help with deriving insights about the model's behavior, detecting faults, and diagnosing the model. Additionally, various other metrics were used to evaluate the trust in those explanations. The work from [9] focuses on identifying the most important features of an AI model predicting Service Level Agreement (SLA) violations in 5G Networks. To achieve this, five different methods were used: SHAP, LIME, PFI, XGBoost, and ELI5. Knowing the ground truth, the authors concluded that all the methods correctly ranked the true root cause of SLA violations in the first two positions. In [10], feature importance explanations of tree-based classifiers for wireless intrusion detection were derived using the SHAP method. The AI classifiers were then simplified through an importance-based feature selection process that resulted in a reduction of the model size by 90%, without performance loss.

Our work will differ from the previous ones in terms of the XAI methods used. We focus on a different set of XAI methods: Kernel SHAP, CERTIFAI, and Anchors. Additionally, we will generate feature importance explanations for the use case of RL-based energy-efficient RRM. Finally, as opposed to the related work, we will simultaneously focus on simplifying the model and improving its understandability.

III. XAI methodology

Our methodology can be split into two stages. The first stage involves generating feature importance explanations of the given AI model using the three XAI methods. The second stage uses the explanations generated to answer various questions and simplify the AI model. The following subsections give a brief overview of these methods and the process of generating feature importance scores through them.

A. Feature Importance explanations

SHAP is a feature importance method used to compute local importance scores for each input feature of an AI model [11]. These scores are called SHAP values, and they quantify the contribution of the features to the model's output. Kernel SHAP values are estimations of Shapley values obtained using a special Linear LIME model [11]. To obtain a global feature importance score for a feature f of an AI model, we calculate the mean of the absolute SHAP values of that feature computed over the data points X_i of a relatively large dataset X from the corresponding domain. This process can be formalized as (1) where $\lambda(f, X_i)$ is the Kernel SHAP value of a feature f computed over X_i, as described in [11].

$$\beta(f) = \frac{\sum_{i=1}^{|X|} |\lambda(f, X_i)|}{|X|}, \tag{1}$$

B. Counterfactuals explanations

CERTIFAI is a contrastive method used to create counterfactuals through a genetic algorithm [12]. A counterfactual c of a data point x is just another data point that has some minimum changes when compared with x, such that an AI model m gives a different output for x and c (i.e., $m(x) \neq m(c)$). In other words, this technique generates instances close to the input that change the model's prediction. It can be used to determine the robustness and sensitivity of the model to its inputs. To compute global feature importance scores using this method, we propose a process similar to the one described in [12]. Firstly, for each data point X_i of a relatively large dataset X from the corresponding domain, generate a set of counterfactuals C_i. Then, for each feature f of the AI model, count how many times the original value from X_i changes across C_i (for all X_i of X). Lastly, this count is normalized, thus producing the final CERTIFAI feature importance score κ, which can be formalized as (2) where $\alpha(a, b) = 1$ if $a \neq b$, or 0 otherwise.

$$\kappa(f) = \frac{\sum_{i=1}^{|X|} \sum_{c \in C_i} \alpha(X_i[f], c[f])}{\sum_{i=1}^{|X|} |C_i|}, \tag{2}$$

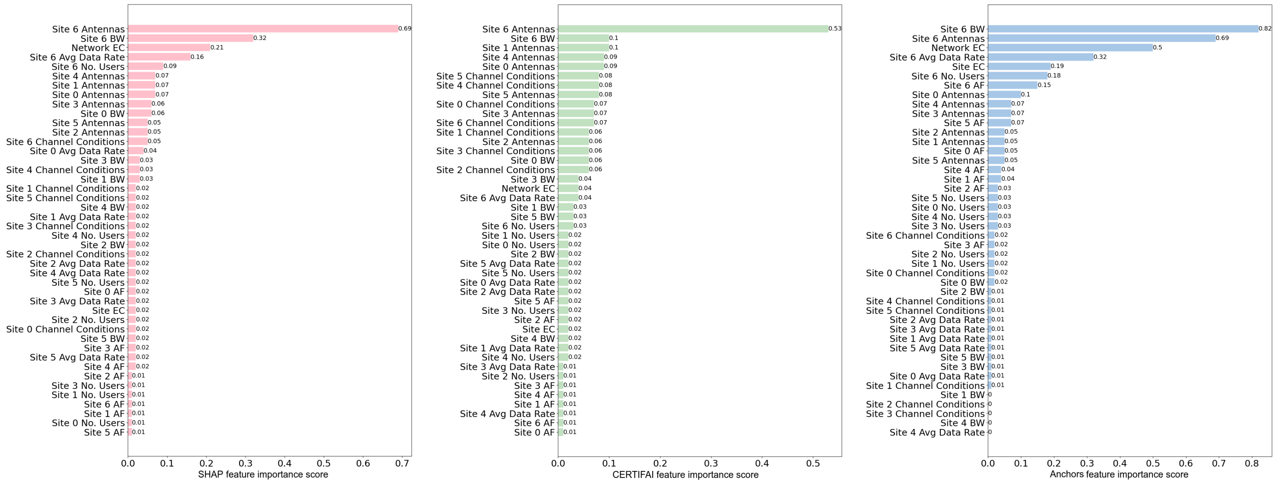
C. Rule-based explanations

Anchors is a model-agnostic knowledge extraction method used to derive IF-THEN rules that describe the reasoning of an AI model in a human-interpretable way [13]. An anchor specifies value constraints for some of the input features of an AI model, along with a specific model prediction that can be expected for a data point x if it satisfies the given constraints. Anchors are built in a bottom-up approach (i.e., predicate by predicate) through a beam-search process based on a KL-LUCB algorithm until a certain anchor precision is met [13]. It is important to observe that Anchors comprise of two key parameters, namely (1) precision, which is the probability of a rule-based prediction matching the model's prediction, and (2) coverage, which is the number of instances or observations that a single rule can explain.

$$Precision \propto \frac{1}{Coverage} \tag{3}$$

In Eq. 3 we observe that, as the number of feature predicates in the anchor increases, the coverage, that is the number of predictions that the model can predict with that rule/anchor goes down. This trade-off between precision and coverage could impact our understandability of the explanations.

In our work, we formulate and use a new Anchors-based feature importance score for a feature f of an AI model. This score is based on the feature's presence within the rules constructed by Anchors. To compute this score, we need to



(a) Kernel SHAP feature importance ranking (b) CERTIFAI feature importance ranking (c) Anchors feature importance ranking

Fig. 1. Feature importance rankings of the RL agent for energy-efficient RRM.

distinguish between three types of rules: (A_i) candidate rules resulting at the end of each except the last iteration of the method, (A_{ii}) candidate rules resulting at the end of the last iteration of the method, and (A_{iii}) the final rule (i.e., the anchor) yielded by the method. Next, we need to count the feature's presence within each type of rule, then normalize, weigh, and sum up the feature presence counts. This process yields a local importance score μ for a feature f, since the procedure of generating an anchor is based on a single data point. To obtain a global importance score χ, we calculate the mean of several local scores computed over the data points X_i of a relatively large dataset X from the corresponding domain. This process can be formalized as follows:

$$\mu(f, X_i) = W_{A_i} \times \frac{\rho(f, A_i \backslash X_i)}{|A_i \backslash X_i|} + \\ + W_{A_{ii}} \times \frac{\rho(f, A_{ii} \backslash X_i)}{|A_{ii} \backslash X_i|} + W_{A_{iii}} \times \frac{\rho(f, A_{iii} \backslash X_i)}{|A_{iii} \backslash X_i|}; \quad (4) \\ \chi(f) = \frac{\sum_{i=1}^{|X|} \mu(f, X_i)}{|X|},$$

where $A_i \backslash X_i$ denotes the set of rules of type A_i generated from instance X_i, $\rho(f, rules\_set)$ is a function counting the presence of feature f within a given set of rules, and W_{A_i}, $W_{A_{ii}}$, $W_{A_{iii}}$ are arbitrarily-decided weights associated to the presence of a feature within each type of rules identified above. Note that the runtime complexity of equation 4 is linear once the Anchor rule sets are generated. Considering that for an instance X_i the precision ω (as defined in [13]) of the three types of rules satisfy $\omega(A_i) \leq \omega(A_{ii}) \leq \omega(A_{iii})$, we suggest that the values of the weights should also satisfy $0 \leq W_{A_i} \leq W_{A_{ii}} \leq W_{A_{iii}} \leq 1$, with $W_{A_i} + W_{A_{ii}} + W_{A_{iii}} = 1$. In this work, we used the following weight values decided through experimentation as they allowed us to achieve the best results, as shown in section V: $W_{A_i} = 0.15, W_{A_{ii}} = 0.25, W_{A_{iii}} = 0.6$. A more detailed description of this Anchors-based feature importance score can be found in [14].

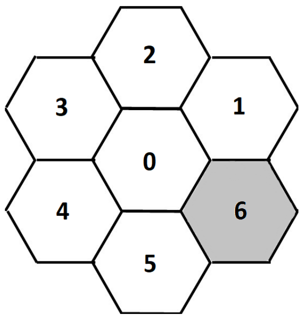
D. Importance-based feature selection

In the second stage of the methodology process, we arrange the obtained feature importance scores into rankings as seen in Fig. 1. By analyzing these rankings, we can identify the most and the least important features for the agent's decisions. These rankings are then used to select features for a simpler AI model. The feature selection strategy is based on repeated feature randomizations and model evaluations. The whole importance-based feature selection process can be summarised as follows. Given a trained AI model m and a test dataset td, an evaluation of m on td is initially performed. Next, given a feature ranking f_r based on the importance scores generated with one of our XAI methods, k least important features are step-by-step randomized within td (we suggest $0.03*|td| \leq k \leq 0.05*|td|$). After each feature randomization step, m is re-evaluated on (the partially randomized) td. When the result of a re-evaluation becomes significantly different than the one of the initial evaluation (e.g. if their difference is bigger than some given ϵ), one can assume that an important-enough feature has been randomized in the current step. Therefore, all features randomized in the previous steps can be removed as they were not important enough to significantly change the result of the initial evaluation of m. This feature selection process is described in more detail in [14].

IV. RL AGENT FOR ENERGY-EFFICIENT RRM

The above-mentioned XAI methods are applied to understand the behavior of an RL agent for energy-efficient RRM. This agent is an evolved version of the work described in [2], [15] in which the learning process is now based on a 3-layer Q-network instead of a Q-table. The goal of the agent is to learn the optimal RRM decisions to reduce both the site and network EC. This is ensured by encapsulating the change in EC caused by the agent's decisions within its reward function.

The network environment is modeled through Python code and resembles the one described in [2]. It contains 7 single-



(a) Deployment of the 7 single-sector sites. Peripheral site no. 6 is highlighted since it is the site at which the RL agent operates

Features monitored at each site
Activity factor (AF)
Channel conditions
No. of active antennas
Utilized bandwidth (BW)
No. of connected users
Avg. data rate

(b) The list of features monitored at each of the 7 single-sector sites

Fig. 2. Various site-related details

sector sites as shown in Fig. 2(a). At each site, we monitor six network variables as listed in Fig. 2(b). A detailed description of these variables is given in [14].

As stated above, the agent is based on a 3-layer neural network. An important aspect of this neural network's architecture is that the hidden layer always contains a variable number of neurons equal to half of the number of input features provided to the agent. This will later improve the degree of complexity reduction of the RL agent that will be achieved through the importance-based feature selection process, as we will see in section V. The non-simplified agent has a total of 44 input features, with 42 of them representing the network variables monitored at the 7 sites, and 2 of them representing the EC of the network and the site at which the RL agent operates. Using the information contained in these features, the RL agent learns to manage the radio resource used at a specific site through a set of 5 decisions. These decisions are (1) increasing or (2) decreasing the number of active antennas at the site, (3) increasing or (4) decreasing the utilized bandwidth at the site by 2 MHz, or (5) not performing any change at all. In this work, we assume that the agent operates at site 6.

Note that the fast-changing nature of these features does not pose a problem for our RL agent or XAI methods. The training of the RL agent is done using static snapshots or states of the environment, and the post-training inference process is fast enough so that the environment does not change in the meantime. Additionally, our XAI methods do not interact with the environment as they only utilize the already-trained RL agent and static environment states.

The data used to train and test the RL agent is collected from the simulated network environment and stored in the agent's replay memory. The validity of this data is ensured by the fact that the network deployment, configuration, behavior, power consumption model, and traffic demand model are designed based on theory, mathematical formulas, and best practices from the field of wireless systems. Detailed information regarding these aspects can be found in [2], [3], [15].

V. Results

The results of our work can be divided into four categories: (A) model understandability, (B) importance-based feature selection, (C) computational resources required by the XAI methods, and (D) Anchors-based inference process.

A. Model understandability

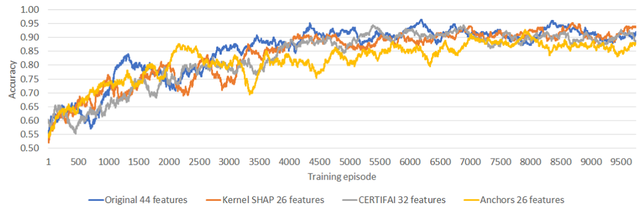
By applying the XAI methods described in section III to the RL agent presented in section IV, we generated the feature importance rankings displayed in Fig. 1. According to these, all three XAI methods agree that the two most important features of our RL agent are the number of active antennas and the utilized bandwidth at the site where the agent operates, i.e., site 6. Additionally, Kernel SHAP and Anchors agree that the third most important feature is the network EC. However, the high importance of these three features might be intuitive, considering that the number of active antennas and the utilized bandwidth is changed by the agent's decisions, and the network EC is included in the reward function.

Nevertheless, the most insightful aspect comes from the bottom of the rankings. As we see, Kernel SHAP and CERTIFAI place the activity factors among the least important features of the RL agent. This is surprising since [2] and [15] show the activity factor of a site is important for achieving an energy-efficient network state. However, in those works, the RL agent uses only the activity factor as input, whereas we consider more granular features such as the number of active antennas, the utilized bandwidth, and the average data rate, which are included in the computation of the activity factor of a site, as explained in [3]. Therefore, we can confirm that granular features are more important for our RL agent when making RRM decisions. Fig. 1 enables us to improve our understanding of our RL agent by grasping the extent to which each input feature influences the decisions of the model.

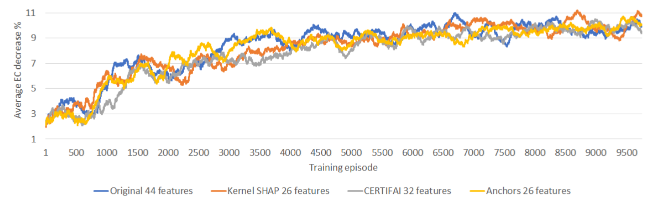
B. Importance-based feature selection: Model complexity reduction and performance evaluation

Based on the rankings presented in Fig. 1, we simplified our RL agent using the feature selection process described in section III. We thereby obtained three different simplified versions of the RL agent, one for each of the XAI methods. To measure the degree of complexity reduction achieved by these simplified agents, we used four different metrics. The exact measurements for each metric are shown in Table I.

Before summarizing the data in Table I, it is important to note that the feature selection process described in section III is variable, meaning that the number of selected features will differ depending on the used feature ranking. Furthermore, the rest of the metrics displayed in Table I are dependent on the number of selected input features. Consequently, since the RL agents simplified based on Kernel SHAP and Anchors feature rankings have the fewest input features, they achieve the highest degree of complexity reduction according to all used metrics. Considering this, in the next paragraph we will refer to these two simplified agents as the ”best models”, and



(a) Accuracy measured during training over a 1-step sliding window of 250 training episodes



(b) Average EC decrease measured during training over a 1-step sliding window of 250 training episodes

Fig. 3. Performance measurements during the training process of our RL agent

TABLE I
COMPLEXITY MEASUREMENTS OF THE THREE XAI METHODS IN THE CONTEXT OF A SINGLE DATA POINT

Metric	Original	Kernel SHAP	CERTIFAI	Anchors
No. of used features	44	26	32	26
Input size (bytes)	352	208	256	208
Q-network computations (feed-forward)	2178	819	1200	819
Q-network parameters	1105	421	613	421

TABLE II
METRICS MEASURED AFTER TRAINING OVER 1000 EPISODES

Metric	Original	Kernel SHAP	CERTIFAI	Anchors
No. of used features	44	26	32	26
Accuracy	90.8%	91.9%	90.9%	91.5%
Avg. EC decrease	9.84%	9.94%	9.98%	9.95%

the agent simplified based on the CERTIFAI feature ranking as the ”worst model”.

In terms of the number of used input features and input size, the best models achieved a reduction of $\sim 41\%$, whereas the worst model achieved a reduction of $\sim 27\%$. In terms of the number of Q-network parameters and computations used in the feed-forward phase, the best models achieved a reduction of $\sim 62\%$, as opposed to the $\sim 45\%$ achieved by the worst model. Regardless, the degree of complexity reduction achieved by the worst model is still noteworthy.

Since our goal was to simplify the RL agent without losing performance, we evaluated and compared the performance of the initial and the simplified agents using two relevant metrics: the accuracy and the average reduction in EC. The accuracy describes the percentage of episodes in which the RL agent managed to decrease both the site and the network EC by the end of the episode. The average EC decrease describes the average percentage of EC decrease at the end of an episode as compared to the beginning of that episode. Using these metrics, we monitored the performance of the agents both during and after the training process.

TABLE III
AVERAGE COMPUTATIONAL RESOURCES USED BY DIFFERENT XAI METHODS

Metric	Kernel SHAP	CERTIFAI	Anchors
Computation time	6 minutes	1 hour	3 hours
Memory usage	550MB	150MB	1700MB

As shown in Fig. 3, the performance of the simplified agents was close to that of the original agent during training, according to both metrics. The small fluctuations observed in those figures are caused by the randomization involved in the training process. Post training, we measured the performance of all agents on the same test dataset and recorded the measurements shown in Table II. As seen, none of the simplified agents have lost any performance when compared to the original agent. In fact, they all showed a slight improvement in performance due to the elimination of less important features that were most likely non-informative and just adding noise to the training data.

C. Computational resources required by the XAI methods

For a more meaningful comparison between the obtained results, we also considered the computational time and the memory usage of each XAI method across several runs and presented the average results in Table III. As we can see, Kernel SHAP was by far the fastest method, whereas Anchors took significantly more time to finish execution. In terms of memory consumption, CERTIFAI was the least demanding, followed by Kernel SHAP and Anchors, with each being 3x more expensive than its predecessor. This makes Anchors the most demanding method in terms of computation time and memory usage. Thus, even though it helped us achieve the same degree of complexity reduction and similar performance of the simplified agents as Kernel SHAP, Anchors is a significantly more expensive method. Hence, taking into account all results presented up to this point, Kernel SHAP is the preferred method for our use case.

D. Anchors-based inference process

In this section, we performed a special set of experiments to analyze the performance obtained by a rule-based inference process. After applying Anchors to our RL agent for computing the feature importance scores, we also obtained

TABLE IV
METRICS MEASURED OVER 1000 TEST EPISODES

Test	Metric	RL inference	Anchors inference
1	Accuracy	90.5%	90.2%
	Avg. EC decrease	9.81%	9.79%
2	Accuracy	92.0%	91.5%
	Avg. EC decrease	9.95%	9.86%
3	Accuracy	89.6%	89.4%
	Avg. EC decrease	9.64%	9.61%

a set of IF-THEN rules that are naturally generated by this method. Thus, for any new data point, we could infer an RRM decision by checking if there is any anchor that applies to this point. If this was the case, then we could infer the decision as suggested by the anchor, else we resort to the default decision of not performing any change in the radio resource utilization. As opposed to the inference process based on our RL agent, the rule-based inference process described above is highly interpretable for humans. This is because IF-THEN rules can easily be followed by humans, especially if they are arranged into a simple decision tree. Following the branches of a decision tree and checking which rules apply is a process that is easily comprehensible by a human, as opposed to the process of applying the multitude of mathematical operations involved by the neural network of a deep RL agent.

As Table IV shows, we applied this rule-based inference process in three different experiments and compared the performance with the RL-based inference process. Using the two metrics described before, we observed that the performance of the rule-based inference process is just slightly lower than the one obtained with RL-based inference. This means that for our use case of energy-efficient RRM, we could replace the less interpretable RL-based inference process with the highly interpretable rule-based inference process with almost no loss in performance. To the best of our knowledge, this is a novel result in the specialized literature.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we proposed an XAI methodology for generating feature importance explanations of any AI model using Kernel SHAP, CERTIFAI, and Anchors. For Anchors, we formulated a new feature importance score based on the feature's presence within the rules constructed by this method. We applied the proposed methodology to increase the understandability and reduce the complexity of an RL agent by generating a new model with a reduced size that requires lower computational and storage resources. Our results show that the anchors-based feature importance score allowed us to achieve the same level of complexity reduction and similar performance of the simplified agents as Kernel SHAP. However, Anchors implied the highest utilization of computational resources compared to the other two methods, making us conclude that Kernel SHAP was the best alternative for our use case and goals. Regardless, if one can afford to pay for the increased cost of utilizing Anchors, then one could use the rules generated to replace the RL-based inference process. However, the performance of the rule-based inference process depends on how well the rules describe a model's behavior. In this use case, we found that the rule-based inference can achieve a performance close to that of the RL-based inference process while having the advantage of being more easily interpretable for humans. In upcoming work, we seek to apply our XAI methodology to RL-based energy-efficient RRM use cases that are closer to real-world scenarios. Such use cases would imply much more input features and possible optimization decisions, and they would allow us to derive more general conclusions about our XAI approach. Lastly, in future work, one could study the impact that varying precision and coverage values have on the number, interpretability, and understandability of the rules generated.

REFERENCES

[1] N. Alliance, "5G White paper," *Next generation mobile networks, white paper*, vol. 1, no. 2015, 2015.

[2] S. K. G. Peesapati, "Energy Efficiency of 5G Radio Access Networks," 2020.

[3] S. K. G. Peesapati, M. Olsson, M. Masoudi, S. Andersson, and C. Cavdar, "An analytical energy performance evaluation methodology for 5G base stations," in *2021 17th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*. IEEE, 2021, pp. 169–174.

[4] F. D. Calabrese, L. Wang, E. Ghadimi, G. Peters, L. Hanzo, and P. Soldati, "Learning radio resource management in RANs: Framework, opportunities, and challenges," *IEEE Communications Magazine*, vol. 56, no. 9, pp. 138–145, 2018.

[5] W. Guo, "Explainable artificial intelligence for 6G: Improving trust between human and machine," *IEEE Communications Magazine*, vol. 58, no. 6, pp. 39–45, 2020.

[6] M. T. Ribeiro, S. Singh, and C. Guestrin, "Model-agnostic interpretability of machine learning," *arXiv preprint arXiv:1606.05386*, 2016.

[7] A. B. Arrieta, N. Díaz-Rodríguez, J. Del Ser, A. Bennetot, S. Tabik, A. Barbado, S. García, S. Gil-López, D. Molina, R. Benjamins *et al.*, "Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI," *Information fusion*, vol. 58, pp. 82–115, 2020.

[8] P. Barnard, I. Macaluso, N. Marchetti, and L. A. DaSilva, "Resource reservation in sliced networks: An explainable artificial intelligence (XAI) approach," in *ICC 2022-IEEE International Conference on Communications*. IEEE, 2022, pp. 1530–1535.

[9] A. Terra, R. Inam, S. Baskaran, P. Batista, I. Burdick, and E. Fersman, "Explainability methods for identifying root-cause of SLA violation prediction in 5G network," in *GLOBECOM 2020-2020 IEEE Global Communications Conference*. IEEE, 2020, pp. 1–7.

[10] S. Bhandari, A. K. Kukreja, A. Lazar, A. Sim, and K. Wu, "Feature selection improves tree-based classification for wireless intrusion detection," in *Proceedings of the 3rd International Workshop on Systems and Network Telemetry and Analytics*, 2020, pp. 19–26.

[11] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," *Advances in neural information processing systems*, vol. 30, 2017.

[12] S. Sharma, J. Henderson, and J. Ghosh, "CERTIFAI: Counterfactual explanations for robustness, transparency, interpretability, and fairness of artificial intelligence models," *arXiv preprint arXiv:1905.07857*, 2019.

[13] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 32, no. 1, 2018.

[14] A.-D. Marcu, "Explainable Artificial Intelligence for Radio Resource Management Systems: A diverse feature importance approach," 2022.

[15] S. K. G. Peesapati, M. Olsson, M. Masoudi, S. Andersson, and C. Cavdar, "Q-learning based radio resource adaptation for improved energy performance of 5G base stations," in *2021 IEEE 32nd Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. IEEE, 2021, pp. 979–984.