CHALMERS
UNIVERSITY OF TECHNOLOGY

Towards Explainable Reinforcement Learning in Optical Networks: The RMSA Use Case

Towards Explainable Reinforcement Learning in Optical Networks: The RMSA Use Case

Omran Ayoub[1,*], Carlos Natalino[2], and Paolo Monti[2]

[1] *Department of Innovative Technologies, University of Applied Sciences of Southern Switzerland, Lugano, Switzerland.*
[2] *Department of Electrical Engineering, Chalmers University of Technology, Chalmers, Sweden.*

**Corresponding author: omran.ayoub@supsi.ch*

Abstract: We propose an approach to extract explanations from a trained reinforcement learning agent. Our analysis over three RMSA environment variations shows how the agent uses the input information, increasing our understanding of its learned policy.

1. Introduction

In recent years, reinforcement learning (RL) has emerged as a powerful tool for addressing complex and dynamic optimization problems in optical networks such as routing and wavelength assignment (RWA) [1], routing, modulation and spectrum assignment (RMSA) [2], and virtual network embedding [3]. The performance benefits achieved by RL demonstrate its potential for application in real-world scenarios. However, similar to other machine learning (ML) algorithms, RL models have a black-box nature that make them hard to explain. Such characteristic makes RL models difficult to trust, and consequently adopt, in real optical network deployments.

For supervised ML models, SHAP (SHapley Additive exPlanations) [4], an eXplainable AI approach, has been widely adopted. SHAP is a method based on cooperative game theory used to quantify the impact of individual features on a model's decision-making process. SHAP values offer insights into the relative importance (influence) of each feature, estimating how they contribute to the model's output. This explainability framework has shown promising properties when applied to quality-of-transmission (QoT) prediction tasks [5].

Recently, explainable RL (XRL) has received increasing attention, driven by the need to interpret and make transparent the decision-making processes of RL models. In the context of optical networks, the notion of XRL remains relatively unexplored. Previous works proposed to interpret and explain the decisions of RL models through the reverse-engineering the network state [6], or the analysis of resource usage (links) in the network [1,7]. However, these studies do not analyze how different features impact the decisions of the RL agent. Therefore, there is still a gap of understanding of the policies learned by RL agents in the context of optical networks. This is critical due to the need of network operators to understand the reasoning behind the RL learned policies prior to its deployment in real networks.

In this work, we aim at leveraging SHAP (SHapley Additive exPlanations) [4] to explain the behavior of RL models applied to the RMSA problem. For this purpose, we propose an approach that uses observations and actions from a trained RL agent to train an ML model in a supervised learning fashion. The resulting ML model is used by SHAP to extract explanations.

Each component of the RMSA problem is solved separately similarly as in [2], with the RL agent solving the routing problem, the modulation format selection being based on the path length, and the spectrum assignment based on the first-fit policy. We analyze three variations of the problem, varying the reward function and the possibility of selecting infeasible actions by the RL agent. We are particularly interested in explaining the important network and lightpath request characteristics that lead the RL model to reject the request. The results allow us to identify which features and which value ranges influence the RL agent towards accepting or rejecting a lightpath request. We observe that by changing the reward function, the RL policy changes the important features that are taken into account when rejecting the requests. Moreover, introducing a mask that prevents the RL model to take infeasible actions makes the importance of features more evenly distributed across the different route options. We believe that the proposed approach can be valuable to increase the trustworthiness of RL models to be deployed in real networks.

2. Problem Description and Proposed Approach

In this work, we assume an elastic optical network (EON) under dynamic network traffic where lightpath requests arrive following a Poisson process. Each lightpath request is processed individually through the solution of the RMSA problem. Lightpath requests have a defined source, destination, and bit rate to be accommodated in the network. We adopt the *DeepRMSA* environment presented in [2] to represent the EON. A RL agent is responsible for interacting with the environment and selecting the route to be used by the request. The modulation format is

decided based on the route length, while the spectrum is selected using the *first-fit* strategy. At the arrival of each lightpath request, the environment exposes an observation used as input to the RL model.

The observation has the following components: *(0)* bit rate, *(1)* source node (*src_node*), *(2)* destination node (*dst_node*), *(3)* starting index for the request if path x is selected (*init_index_x*), *(4)* number of required slots if path x is selected (*req_slots_x*), *(5)* number of free slots in the first free spectrum block if path x is selected (*free_sl_block_x*), *(6)* total number of free slots across path x (*free_sl_path_x*), and *(7)* average number of slots across all free spectrum blocks across path x (*avg_sl_path_x*). Components *3-7* are defined for each of the possible k paths, with x=0..$k-1$. In our case, we adopt k=5 shortest paths, resulting in a total of 28 features. The action space is composed of $k+1$ alternatives corresponding to selecting one of the k shortest paths or rejecting the request. The reward function assumes value 1 if the request is successfully provisioned, 0 otherwise.

We devise three variations of the environment. The first one, referred to as *infeasible-allowed*, adopts the modeling from [2] described above. The second one, referred to as *infeasible-penalized*, has a different reward function. It assumes a value of 1 if a request is successfully provisioned, and -1 if the request is not provisioned. In both cases, the agent is allowed to select any of the $k+1$ available actions, even if an action may not be feasible due to lack of resources. Selecting one of these actions will lead to the rejection of the request. The third option, referred to as *infeasible-masked*, uses the same action space and reward function as the *infeasible-penalized* variation but adopts a masking strategy that prevents the agent from selecting infeasible actions. Action masking has been shown to improve the performance of RL agents in routing tasks [1, 8].

Our proposed approach consists of four steps. *Step 1: Training of the RL model.* We train the RL models with an input set of 28 features representing the network state and the lightpath requests. As output we produce a provisioning action, with the objective of maximizing the long-term reward. The specific algorithm for training the agent can be any of the vast RL training strategies for discrete actions. *Step 2: Testing of RL model and data collection.* We test the RL model by letting it interact with the environment and collecting the data exchanged between agent and environment (i.e., input features and actions). The data collected is used to build a dataset where the input features are associated with the respective actions as ground-truth labels. *Step 3: Training a Classifier.* We train a classifier (or a selection of classifiers) on the collected data in a supervised learning manner with the aim of characterizing the behavior of the RL model. The classifier achieving the highest performance in terms of F1-score or accuracy for each of the RL models is considered for the next step. *Step 4: Explaining model using SHAP.* In the last step, we use SHAP to explain the behavior of the trained classifier. We extract explanations of all data points in the test set and obtain the SHAP values (i.e., values of feature impact on the model's decisions) for all features. Finally, we use the explanations to uncover the properties of the policy learned by the RL agent and assess its reasoning.

3. Numerical Evaluations

The three DeepRMSA environment variations described in the previous section are implemented using the Optical RL-Gym [9]. We train one proximal policy optimization (PPO) agent for each of the environments using the same optical network scenario as in [2], i.e., the NSF network topology, with 320 frequency slots in each link, and a non-uniform traffic profile.

Let us first discuss how the variations in the environments impact the performance of the RL agent in terms of episode reward. Moving from the *infeasible-allowed* to the *infeasible-penalized* environment reduces the episode reward by 60%, due to the introduction of a penalty (-1) when the latter rejects requests. When action masking is introduced, we observe an increase of 13% in the episode reward compared to the case without masks, a similar outcome as the ones reported for RWA in [1]. This is explained by the fact that infeasible actions, i.e., infeasible paths due to lack of resources, are now masked out and cannot be selected.

Now we move our attention to the performance of the classifier trained based on the RL agent actions. We evaluated various classifiers such as Random Forest (RF), eXtreme Gradient Boosting (XGB), Light Gradient

Fig. 1: Min-max cross-validation performance of the RF classifier.

Model	Infeasible Allowed	Infeasible Penalized	Infeasible Masked
F1-score	0.75-0.85	0.74-0.84	0.77-0.84
Accuracy	0.84-0.94	0.77-0.91	0.84-0.94
P (Class 5)	0.75-1.00	0.67-0.9	0.75-1.00

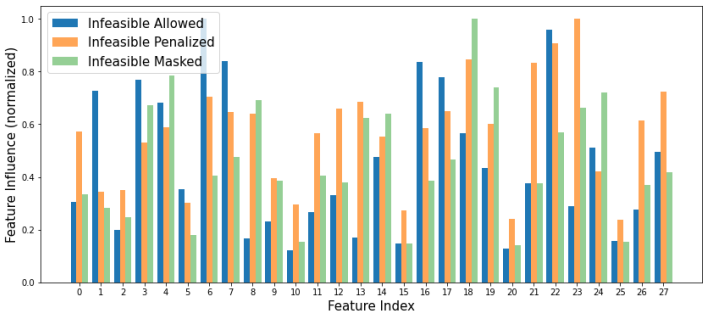
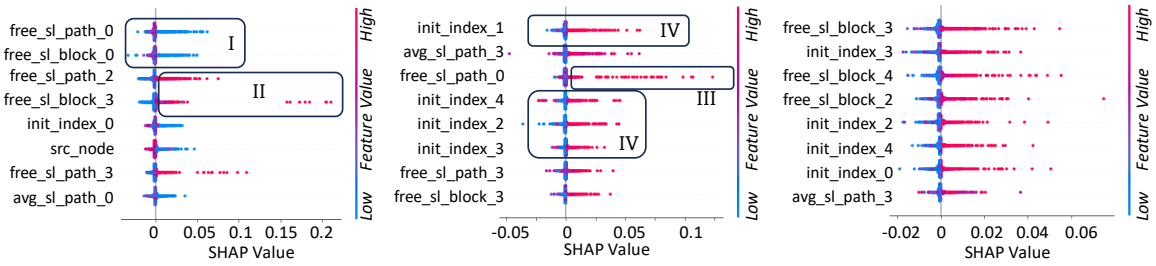


Fig. 2: Normalized SHAP values for all features for each of the models.



(a) Infeasible-Allowed (b) Infeasible-Penalized (c) Infeasible-Masked

Fig. 3: SHAP summary plot (showing top 8 features) for each of the models for rejecting a request

Boosting (LGB) over the dataset collected from the RL agent. We perform hyperparameter tuning using grid search and then test each of the classifiers considering an 80%-20% split for training-testing. An RF classifier achieved the best performance in imitating the behavior of all RL models. Table 1 reports the performance of the classifiers in terms of ranges of accuracy, average F1-score and precision for class 0 (selecting the shortest path) and class 5 (rejecting the request), which are the classes under investigation in our study. The classification results demonstrate that the classifiers can closely imitate the actions of the RL models.

Fig. 2 shows the normalized (with respect to the maximum) feature influence of the 28 features for each of the RL models. We can observe that the feature influence differs significantly among the models. This suggests that the environment variations lead to models that prioritize features differently. For instance, feature 6 (total number of free slots in the shortest path) is the most important feature for *infeasible-allowed*, while it is of relatively low importance for *infeasible-masked*.

We now turn our attention to analyzing SHAP's summary plot focusing on action 6 (rejecting a request) for *infeasible-allowed*, *infeasible-penalized* and *infeasible-masked* in Figs. 3(a), (b) and (c), respectively. A SHAP summary plot for a specific action (class) provides a concise visual representation of the impact and direction of each feature's contribution (positively, pushing the prediction higher for that class or negatively, pushing prediction lower). The y-axis represents the features, ranked in order of their absolute SHAP values. The x-axis displays the SHAP values or the magnitude and direction of each feature's contribution.

Fig. 3(a) shows that, for *infeasible-allowed*, low values in the number of available slots influence the model towards rejecting the request (inset I), which is intuitive. However, inset II shows that high values of free slots in the third or fourth shortest paths also influence the model towards rejecting the request. This suggests that *infeasible-allowed* suggests rejecting a request even if capacity is available on longer paths. Fig. 3(b) indicates that the *infeasible-penalized* leads to a different decisional process. Inset III shows that the model may reject a request even if the number of free slots in the shortest path is high. Moreover, inset IV highlights the fact that high initial indices for the request also influence the model towards rejection. This suggests that the *infeasible-penalized* environment leads to a model that is more conservative with respect to using resources. Similarly, the *infeasible-masked* variation uses mainly the initial indices across the paths and the number of free slots to decide on the rejection of lightpath requests.

Acknowledgment: This work was partially supported by Vetenskapsrådet (2022-04798).

References

1. J. W. Nevin *et al.*, "Techniques for applying reinforcement learning to routing and wavelength assignment problems in optical fiber communication networks," J. Opt. Commun. Netw. **14**, 733–748 (2022). DOI: 10.1364/JOCN.460629.
2. X. Chen *et al.*, "DeepRMSA: A deep reinforcement learning framework for routing, modulation and spectrum assignment in elastic optical networks," J Light. Technol. **37**, 4155–4163 (2019). DOI: 10.1109/JLT.2019.2923615.
3. M. Dolati, S. B. Hassanpour, M. Ghaderi, and A. Khonsari, "DeepViNE: Virtual network embedding with deep reinforcement learning," in *IEEE INFOCOM Workshops,* (2019), pp. 879–885. DOI: 10.1109/INFCOMW.2019.8845171.
4. S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Advances in neural information processing systems,* vol. 30 (2017), p. 4768–4777.
5. O. Ayoub *et al.*, "Towards explainable artificial intelligence in optical networks: the use case of lightpath QoT estimation," J. Opt. Commun. Netw. **15**, A26–A38 (2023). DOI: 10.1364/JOCN.470812.
6. J. Suarez-Varela *et al.*, "Routing in optical transport networks with deep reinforcement learning," J. Opt. Commun. Netw. **11**, 547–558 (2019). DOI: 10.1364/JOCN.11.000547.
7. S. Nallaperuma *et al.*, "Interpreting multi-objective reinforcement learning for routing and wavelength assignment in optical networks," J. Opt. Commun. Netw. **15**, 497–506 (2023). DOI: 10.1364/JOCN.483733.
8. M. Shimoda and T. Tanaka, "Mask RSA: End-to-end reinforcement learning-based routing and spectrum assignment in elastic optical networks," in *Proc. of ECOC,* (2021), p. Th1E.4. DOI: 10.1109/ECOC52684.2021.9606169.
9. C. Natalino *et al.*, "The Optical RL-Gym: An open-source toolkit for applying reinforcement learning in optical networks," in *Proc. of ICTON,* (2020). DOI: 10.1109/ICTON51198.2020.9203239.