Experimental Validation of MLOps for Automated Service Provisioning in Optical Networks

Carlos Hernández-Chulde, Ramon Casellas, Ricardo Martínez
Centre Tecnològic de Telecomunicacions de Catalunya (CTTC/CERCA)
Castelldefels (Barcelona), Spain
carlos.hernandez@cttc.es

***Abstract*—The increasing complexity and stringent demands of modern communications require advanced automation since traditional methods cannot effectively address the dynamic nature of network operations. This demonstration showcases a Machine Learning Operations (MLOps) framework tailored for optical networks. It highlights the deployment and operation of deep reinforcement learning (DRL) models to tackle complex, dynamic challenges in optical networks by presenting practical use cases. Through an experimental setup, it illustrates the end-to-end lifecycle management of DRL agents integrated into an SDN control plane. The demonstration spans data collection, model training, deployment, and continuous monitoring, showcasing the unique benefits and challenges of intelligent optical network operations. The demonstration provides practical insights into deploying intelligent network functions and emphasizes the critical role of MLOps in achieving zero-touch management for future optical infrastructures.**

***Index Terms*—Machine Learning Operations, Elastic Optical Networks, Deep Reinforcement Learning**

I. Introduction

The evolving landscape of the upcoming 6G networks necessitate a paradigm shift towards full automation without human intervention, driven extensively by Artificial Intelligence (AI) and Machine Learning (ML). This transition enables networks to achieve self-optimizing and autonomous operations, addressing the complexity and diverse objectives, such as enhanced Quality of Service (QoS) and Quality of Experience (QoE), infrastructure optimization, and scalable resource utilization [1].

Deploying ML models in complex network environments poses challenges related to managing the entire lifecycle of model creation, deployment, and operation. MLOps (Machine Learning Operations) has emerged as a critical paradigm to address these issues by providing a set of practices that combines ML, DevOps, and data engineering to deploy and maintain reliable and efficient ML models reliably in production [2], [3]. A robust MLOps pipeline streamlines data collection, model generation and deployment, while enables automation, and continuous integration and delivery (CI/CD) [3].

Work funded by the European Commission (EC) through the HEU SEASON (101096120) project, the Ministerio de Asuntos Económicos y Transformación Digital and the European Union- Next GenerationEU in the frameworks of the Plan de Recuperación, Transformación y Resiliencia and of the Mecanismo de Recuperación y Resiliencia through UNICO-5G I+D 6G-OPTRAN under references TSI-063000-2021-22, TSI-063000-2021-23, TSI-063000-2021-115 MCIN/AEI/10.13039/501100011033/FEDER/UE and by ERDF A way of making Europe through the Spanish RELAMPAGO (PID2021-127916OBI00) project.

While existing ML solutions may excel in data engineering or ML model generation, comprehensive pipelines encompassing the entire ML lifecycle are still underdeveloped for optical networks. MLOps brings significant benefits to optical networks by streamlining the entire ML lifecycle, from data collection and model development to deployment, which helps mitigate the inherent complexity and reduces inefficiencies. It supports scalability and automation, enabling CI/CD, and boosts performance and reliability through agile and traceable model development that facilitates real-time analysis and robust data handling. Ultimately, MLOps is essential for managing the dynamic nature of modern optical network infrastructures and plays a critical role in achieving zero-touch management.

However, deploying ML models, especially Deep Reinforcement Learning (DRL) agents carries inherent risks due to exploratory behaviors and minor implementation details can significantly impact performance, making reproducibility challenging. To mitigate these risks and challenges, the use of Digital Twins (DTs) is essential, to provide a safe, controllable, and reliable simulation environment for training and testing DRL agents before deployment in production networks [4]. MLOps ensures reproducibility through meticulous training tracking and versioning, and enables autonomous model selection and validation before deployment. This approach allows for the effective management of the end-to-end lifecycle of DRL agents and their integration into SDN control planes, ensuring enhanced network performance and reliability.

This demonstration presents a practical MLOps framework for DRL-driven automation in optical networks, covering the following critical applications: Latency-aware Routing and Spectrum Assignment (RSA), QoT-aware Resource Allocation in Multi-Band Elastic Optical Networks (MB-EONs), and Energy-aware Routing, Modulation, and Spectrum Assignment (EA-RMSA). The demonstration will showcase the potential of advanced MLOps pipelines in optimizing ML deployment within complex optical network environments.

II. MLOps Framework for DRL-Aided Optical Networks

Our proposed framework employs an MLOps pipeline centered around MLflow [5], designed to manage the complexity of building, training, deploying, and maintaining ML models. This framework adheres to core principles such as:

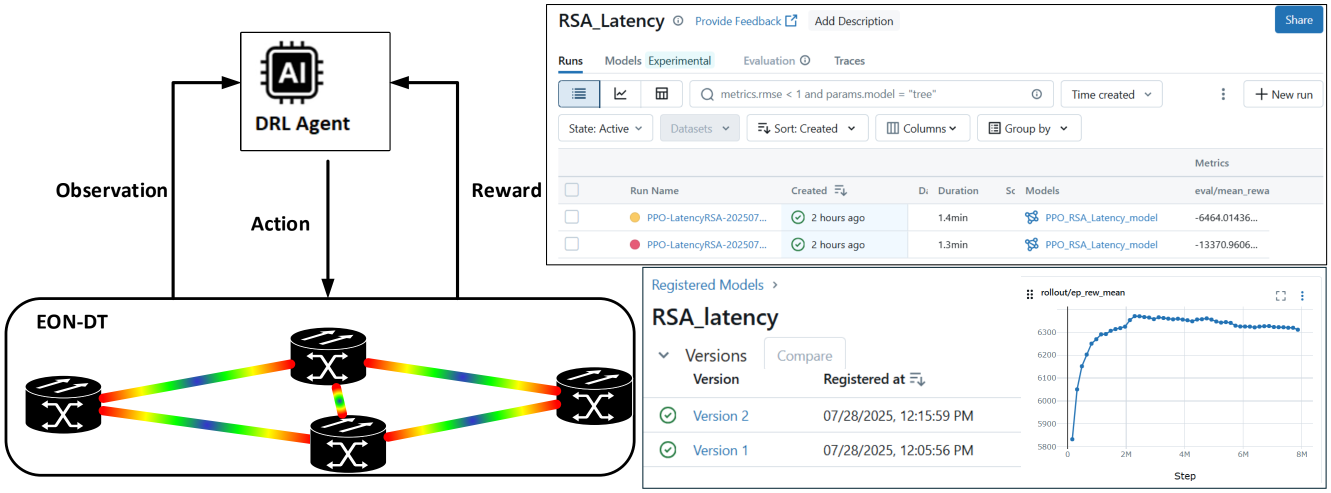

Fig. 1. MLflow User Interface: Experiment tracking and Model registry

CI/CD Automation: Automating the ML pipeline from data to model for continuous training, and CI/CD for ML applications, ensuring rapid delivery while maintaining quality.

Reproducibility and Versioning: Ensuring versioning of data, models with ML metadata tracking/logging, and code for robust and reproducible ML systems.

Continuous Monitoring and Feedback Loops: Tracking model serving performance and network KPIs to enable re-training and continuous improvement.

Digital Twins for Safe Learning: Utilizing DTs to provide a controlled, reliable, and easily accessible simulation environment for safe DRL agent training and exploration, mitigating risks in production network environments.

A. MLflow Implementation for DRL Lifecycle Management

The MLflow's core components leveraged in our demonstration are:

MLflow Tracking: This core component is used to log every aspect of DRL experiments, including parameters, metrics, and artifacts. For DRL agents, this involves meticulously tracking cumulative reward curves over episodes, detailed configurations of the DRL environment (state, action space), and hyperparameters of the Deep Neural Networks (DNNs) that constitute the agent's policy. This enables comparing different reward function designs and DRL algorithmic choices.

MLflow Projects: To ensure reproducibility, our DRL code and environment specifications (e.g., Python dependencies, gym environments) are packaged using MLflow Projects, allowing others to easily run and reproduce experiments.

MLflow Models: After training, DRL policies are standardized and packaged as MLflow Models. This facilitates versioning and registration in the Model Registry. Models are tagged with metadata for traceability.

Model Deployment: Trained DRL models (policies) are deployed to production environments via MLflow's REST API integrated into containerized environments. Post-deployment, continuous monitoring of model performance and logging of predictions are enabled via MLflow.

Figure 1 shows the MLflow User Interface that provides centralized visualization and management capabilities for the ML model lifecycle, particularly through its Experiments Tracking and Model Registry components.

III. Demo Settings and Experimental Setup

To validate the MLOps framework and its DRL agents, we consider an emulated EON setup. In this scenario, a DRL agent is integrated with an ML-assisted Path Computation Element (PCE). During training, the DRL agent interacts with a DT to learn to make optimized resource allocation decisions based on obtained rewards, adapting autonomously to network conditions and dynamic service requirements. The trained DRL agent (ML model) is deployed as an MLflow inference server with REST endpoints. The PCE, externalized from the SDN controller, builds the input for the ML model and sends an inference request to the MLflow inference server. Then, this server sends back the result of the inference. Communication between the SDN controller, PCE and MLflow inference server occurs via standard REST APIs (e.g., API based on ONF Transport API Yang data models). In short, the SDN controller processes connectivity service requests, delegates path computation to the PCE, and then receives the computed path and resource allocation details (links, selected spectrum) to configure network elements. This architecture supports dynamic, automated interaction for service provisioning.

A. Use Cases:

The MLOps framework will address the following use cases:

Latency-aware RSA: A scenario simulating dynamic connection requests will be shown. The demo will visualize the DRL agent's path and spectrum selection in C-band, and illustrate the impact on latency requirements [6].

QoT-aware RMBSA: The demonstration will involve dynamic traffic requests in a multi-band EON (C, L, S bands). The DRL agent's decisions on route, modulation, band, and spectrum allocation, considering QoT estimations (e.g., GSNR) and spectrum usage across different bands will be presented [7].

Energy-aware RMSA: A service provisioning scenario focused on minimizing energy consumption will be presented.

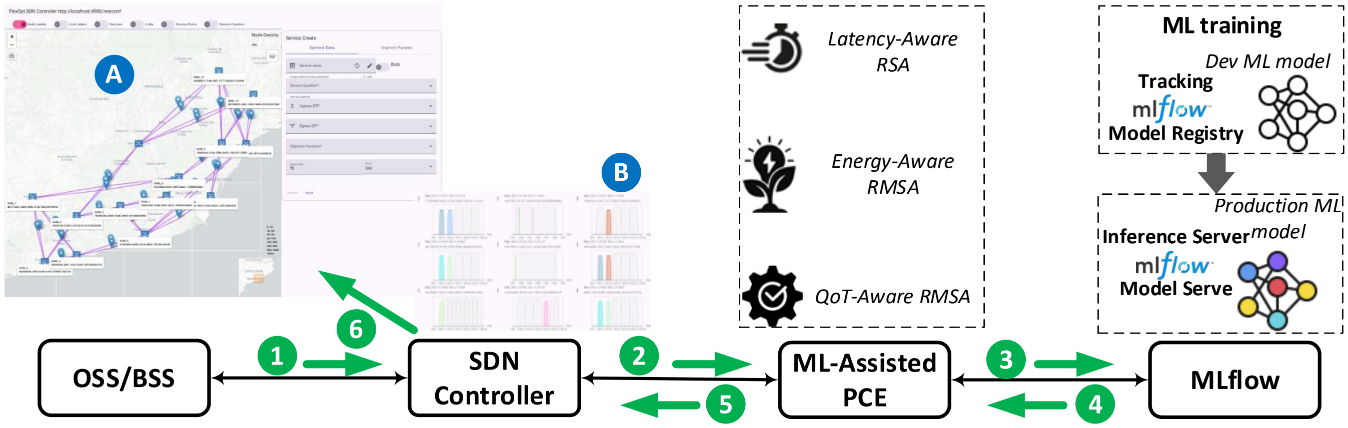

Fig. 2. MLflow Implementation for DRL Lifecycle Management

The demo will show the DRL agent's preference for reusing active network elements and selecting lower-power modulations [8].

IV. Audience Experience and Visual Outcomes

The demonstration aims to provide a clear and compelling illustration of the MLOps framework's effectiveness in implementing DRL in optical networks, as shown in Fig. 2. The audience will observe intelligent automation in action through a structured flow and dynamic visual displays.

A. Interactive Demo Flow

Service Request Initiation: A user will initiate a new connectivity service request via a simplified graphical interface or a REST API call, specifying bandwidth and latency requirements (Step 1).

MLOps Orchestration in Action: The request will trigger the SDN controller, which then queries the external PCE (Step 2). The PCE sends an inference request to the MLflow inference server (Step 3). The underlying MLOps framework facilitates this interaction, ensuring an optimized and updated DRL model to meet specific objectives, such as latency, QoT, or energy.

Real-time DRL Decision: The DRL agent (ML model) computes the optimized resource allocation decision (i.e., node, links, modulation format and spectrum). This decision will be returned to the PCE (Step 4) and ultimately to the SDN controller (Step 5).

Network Configuration: The SDN controller then configures the underlying optical network elements to establish the lightpath (Step 6).

B. Live Service Provisioning Visualization

A network topology map will show the dynamically selected optical path (see Fig. 2 A) and the allocated spectrum for the provisioned services (see Fig. 2 B). This visualization can illustrate how DRL makes optimized decisions, especially under congested scenarios.

C. MLOps Lifecycle Monitoring

While a full re-training cycle may be too long for a live demo, a simplified dashboard, as shown in Fig. 1, could briefly illustrate:

Model Performance Trends: Show a historical view of the DRL model's performance metrics (e.g., achieved reward, accuracy) over time, indicating stability or potential degradation that would trigger re-training.

Model Registry Snapshot: A simple interface displaying the versioning of DRL models and their associated metadata (e.g., training date, hyperparameters, performance metrics), showcasing the reproducibility aspect of MLOps.

V. Conclusion

This demonstration provides strong evidence for the practical effectiveness of deploying an MLOps framework to manage DRL-based network intelligence in optical networks. The integration of DRL agents with SDN control planes via a robust MLOps architecture paves the way for more intelligent, efficient, and reliable network operations, moving towards zero-touch management in 6G systems. Future research will focus on extending the framework to handle more complex scenarios, such as service function chaining, and improving scalability and adaptability to evolving network architectures.

References

[1] F. Rezazadeh *et al.*, "SliceOps: Explainable MLOps for Streamlined Automation-Native 6G Networks," *IEEE Wireless Communications*, vol. 31, no. 5, pp. 224–230, Oct. 2024.

[2] A. Gopikrishnan *et al.*, "Machine Learning Pipeline For Anomaly Detection In Next Generation Networks," in *2024 IEEE Conference on Standards for Communications and Networking (CSCN)*, Nov. 2024.

[3] D. Kreuzberger *et al.*, "Machine Learning Operations (MLOps): Overview, Definition, and Architecture," *IEEE Access*, vol. 11, pp. 31 866–31 879, 2023.

[4] P. Li *et al.*, "RLOps: Development Life-Cycle of Reinforcement Learning Aided Open RAN," *IEEE Access*, vol. 10, pp. 113 808–113 826, 2022.

[5] MLflow. [Online]. Available: https://www.mlflow.org/

[6] C. Hernandez-Chulde *et al.*, "Experimental evaluation of a latency-aware routing and spectrum assignment mechanism based on deep reinforcement learning," *J. Opt. Commun. Netw.*, vol. 15, no. 11, pp. 925–937, Nov. 2023.

[7] C. Hernández-Chulde *et al.*, "Demonstration of Autonomous DRL-Based QoT-Aware Provisioning in Multi-Band EONs," in *2025 25th Int. Conf. on Transparent Optical Networks (ICTON)*, 2025, pp. 1–3.

[8] R. Martínez *et al.*, "Autonomous Control Operations for Energy-Efficient Packet Optical Networks," in *2024 24th Intet. Conf. on Transparent Optical Networks (ICTON)*, 2024, pp. 1–4.