This is the author's accepted version of the article. The final version published by IEEE is R. Soundrarajan, C. Fiandrino, M. Polese, Salvatore D'Oro, L. Bonati, and T. Melodia, "On AI Verification in Open RAN," , IEEE COMMAG, 2025, pp. TBD, doi: TBD.

On AI Verification in Open RAN

Rahul Soundrarajan, Claudio Fiandrino, *Member, IEEE,* Michele Polese, *Member, IEEE,* Salvatore D'Oro, *Member, IEEE,* Leonardo Bonati, *Member, IEEE,* and Tommaso Melodia, *Fellow, IEEE*

***Abstract*—Open RAN introduces a flexible, cloud-based architecture for the Radio Access Network (RAN), enabling Artificial Intelligence (AI)/Machine Learning (ML)-driven automation across heterogeneous, multi-vendor deployments. While EXplainable Artificial Intelligence (XAI) helps mitigate the opacity of AI models, explainability alone does not guarantee reliable network operations. In this article, we propose a lightweight verification approach based on interpretable models to validate the behavior of Deep Reinforcement Learning (DRL) agents for RAN slicing and scheduling in Open RAN. Specifically, we use Decision Tree (DT)-based verifiers to perform near-real-time consistency checks at runtime, which would be otherwise unfeasible with computationally expensive state-of-the-art verifiers. We analyze the landscape of XAI and AI verification, propose a scalable architectural integration, and demonstrate feasibility with a DT-based slice-verifier. We also outline future challenges to ensure trustworthy AI adoption in Open RAN.**

***Index Terms*—Open RAN, AI verification, XAI.**

I. INTRODUCTION

RECENT years have witnessed growing interest in making the Radio Access Network (RAN) more flexible and programmable. The Open RAN vision seeks to transform closed, hardware-bound architectures into virtualized, software-defined systems [1]. The O-RAN ALLIANCE specifications operationalize this vision through an architecture composed of interoperable, multi-vendor components connected by standardized interfaces and the AI-RAN Alliance further promote the integration of Artificial Intelligence (AI)/Machine Learning (ML) into the RAN ecosystem, fostering standardization and adoption of intelligent control mechanisms.

AI/ML technologies are emerging as key enablers in this ecosystem, powering closed-loop control mechanisms that optimize utility metrics (e.g., throughput) through trial-and-error. According to a 2024 SNS Telecom & IT report [2], investments in Open RAN automation are expected to grow by 125 % by 2027, reaching nearly $700 million. Deep Reinforcement Learning (DRL) techniques are increasingly used for RAN resource allocation, handover, load balancing [3], and misconfiguration handling [4], thanks to their adaptability in distributed and dynamic environments. However, these models often operate as closed boxes, limiting interpretability and reducing operator trust. Researchers have adopted EXplainable Artificial Intelligence (XAI) techniques [5], yet explainability without verifiability does not guarantee alignment with operational goals like Service Level Agreement (SLA) for latency and throughput, strict slice isolation, reliability and packet-loss bounds, fairness across users, and energy or cost budgets.

AI verification complements XAI by checking whether AI models adhere to specified behaviors and perform reliably under expected conditions [6], [7] (e.g., in the absence of adversarial inputs). Through formal methods and testing, verification tackles safety, fairness, and robustness challenges that explanations alone cannot address. Unlike traditional approaches as model checking [8], exhaustive state-space exploration [6], or SMT-based verification [9], which are often computationally prohibitive, we introduce a lightweight, pragmatic verification strategy based on interpretable models like Decision Trees (DTs). Our approach provides near-real-time feedback and is suitable for deployment alongside DRL-based xApps, such as those responsible for RAN slicing and scheduling.

In summary, this work addresses a key gap in Open RAN by integrating efficient verification into

---

This work is partially supported by bRAIN project PID2021-128250NB-I00 funded by MCIN/ AEI /10.13039/501100011033/ and the European Union ERDF "A way of making Europe"; C. Fiandrino is a Ramón y Cajal awardee (RYC2022-036375-I), funded by MCIU/AEI/10.13039/501100011033 and the ESF+. This work was also partially supported by the O-RAN ALLIANCE, and by the U.S. National Science Foundation under grant CNS-2112471.

R. Soundrarajan is with Tejas Networks, Bengaluru, India. E-mail: {rahulsound}@mail.com.

C. Fiandrino is with IMDEA Networks Institute, Madrid, Spain. E-mail: {firstname.lastname}@imdea.org.

M. Polese, S. D'Oro, L. Bonati, and T. Melodia are with Northeastern University, Boston, MA, USA. E-mail: {m.polese, l.bonati, s.doro, t.melodia}@northeastern.edu

AI/ML-driven RAN control. Our contributions are threefold. First, we provide a critical overview of the roles of XAI and AI verification, clarifying their interplay in the Open RAN context. Second, we introduce a system-level architectural mapping to embed verification into the O-RAN AI/ML lifecycle, identifying where telemetry and decisions can be introspected. Third, we present a use case where a slice-verifier based on DT models verifies DRL-driven RAN slicing and scheduling, demonstrating its feasibility within RIC timing constraints. Finally, we outline future directions for scalable and trustworthy AI verification in Open RAN environments.

Overall, this paper aims to stimulate the research at the intersection of XAI, AI verification and networking to continue improving AI robustness for its adoption in Open RAN.

II. AI VERIFICATION AND XAI

Verifying learning-based systems means checking whether a trained model satisfies specific behavioral or robustness criteria—such as resilience to adversarial inputs, tolerance to missing features, or generalization to unseen data. Traditional verification approaches often rely on formal tools like mixed-integer linear programming or logic-based solvers, but these lack scalability and generality across diverse scenarios [8], [7]. By contrast, explainability aims to make model behavior human understandable using methods such as decision rules, counterfactuals, or knowledge graphs [10].

Verification and explainability are closely linked in learning-based systems, much like decision and optimization in computational complexity. Recent work shows that verification challenges (e.g., adversarial robustness) can inform explainability, and vice versa [11]. Unifying these concepts offers deeper insights into AI decision-making and helps prevent errors before deployment.

A. Applications and Benefits of XAI

XAI in Brief. Explainability is key to building trust in AI systems. XAI techniques fall into three main categories: post-hoc, counterfactual, and model-inherent. Post-hoc methods such as Local Interpretable Model-agnostic Explanations (LIME), SHapely Additive exPlanations (SHAP), LayeR-wise backPropagation (LRP), and GRAD-CAM analyze models after training and can be applied to any closed-box model. Surrogate models like DTs also belong to this group. Counterfactual approaches explain decisions by showing how small input changes could alter predictions, offering insights into decision boundaries. Model-inherent techniques use self-interpretable models (e.g., DTs) or design models with interpretability embedded in the training process. Coupling explainability with robustness is key for telecom resource management and explainable models should be designed to operate within closed-loop automation frameworks such as Zero-touch Service Management (ZSM) [12].

Challenges of XAI in the Open RAN. The Open RAN framework lays the path forward for closed-loop control, with the RAN exposing context and telemetry to controllers and third-party AI/ML applications that dynamically tune and optimize the network. In this context, the development of efficient control mechanisms remains an open research area. As the industry transitions toward more intelligent control, there is a pressing need for robust, reliable, and deployable ML solutions specifically tailored to wireless networks. However, the existing solutions often rely on a combination of heuristics and AI/ML techniques. These include optimizing training methods for dynamic online systems, developing DRL solutions that generalize well across diverse deployments, and exploring the potential of generative AI in the Open RAN context [1].

Unfortunately, achieving explainability in systems like cellular networks with dynamic control and complex input/output relationships is complex:

- The RAN is a non-stationary and dynamic environment with varying operational conditions such as channel propagation characteristics and traffic profiles. As these conditions are very hard to model, the generalizability of model-free AI/ML techniques is key for effective solutions. At the same time, this makes the generation of explanations that are informative and consistent with the current dynamics of the system challenging.
- For the explanations to be effective, the computational complexity of the XAI techniques must be kept under control. In the RAN, operations occur at timescales that stem from few milliseconds to several seconds. However, techniques like SHAP could take up to several hours to produce global explanations [5].

Benefits of XAI in the Open RAN. In the context of Open RAN, specific benefits of XAI include:

• *Troubleshooting and monitoring* of the AI/ML models to prevent misconfigurations and help with conflict mitigation. The Open RAN ecosystem is heterogeneous with hardware and software from diverse manufacturers and different radio access technologies [4]. Therefore, automating network operations via AI/ML could lead to misconfigurations that are hard to troubleshoot and resolve without deep knowledge about the AI/ML decisions.

• *Intent-based networking* aims at enabling simplified and agile network management where complex configurations are translated into high-level intents such as guarantee Ultra-Reliable and Low Latency Communications (URLLC) 95-th-percentile latency $\leq x$ ms. In this context, XAI can shed light whether the models responsible for network configuration are consistently enforcing the given intents [5].

B. The Current Landscape of AI Verification

Formal methods have been successfully applied to verify classical software systems and only in recent years researchers have started to investigate their application to AI/ML [8], [6]. Vis-a-vis with the classical software systems, AI/ML systems pose specific challenges. Beyond being inherently complex and difficult to understand, AI/ML solutions for cellular networks are also significantly affected by the unpredictable nature of wireless channels and struggle to perform well on data different from what they were trained on.

The landscape of AI verification includes *probabilistic* and *non-probabilistic* methods [8], [9]. The latter were originally conceived for software with deterministic input-output nature and strive to prove that the outputs of an AI model satisfy given criteria for all inputs in a given input space. Such techniques are Satisfiability Modulo Theories (SMT), Mixed Integer Linear Programming (MILP), reachability analysis. Although these techniques are very effective, they scale poorly [8]. By contrast, probabilistic techniques are tailored to AI systems and aim either to find upper bounds on the probability of respecting/violating a given criteria or to estimate the confidence of the output of the model given the statistical properties of the input. These techniques are more scalable at the expense of soundness in executing verification operations [9], [7].

The class of AI/ML models that are of interest for the Open RAN encompasses deep learning algorithms for regression and classification tasks, and DRL [1]. Verifying regressors and classifiers is an important task, but it is often of a different scope than verifying DRL agents deployed for RAN control. The latter directly impact network operations while the former classes of models require additional processing before acting upon network mechanisms. Verifying the operation of DRL involves the assessment of the degree of safety of the actions that the agent takes, i.e., if there is no harm on the state and whether there exist variations in the effect of one action enforced at different times. The verification process for DRL agents is complex because the output of one decision becomes the input for the next, creating a cumulative effect which adds exponential complexity to verification tools as the number of time steps increases [13].

III. AI VERIFICATION IN THE OPEN RAN

Verifying AI behavior in Open RAN is essential to ensure that autonomous decisions made by AI/ML models align with operational goals like SLA assurance. Within the O-RAN architecture, verification must occur continuously and transparently, without disrupting ongoing network operations. This section outlines how AI verification integrates into the existing O-RAN workflow, and where verification modules can be deployed.

AI and ML play a central role in the O-RAN architecture, which has been designed to support the AI/ML lifecycle from data collection and training to deployment and inference or control based on live data from the network, as shown in Fig. 1. This is facilitated by a set of open interfaces, which expose data, telemetry, and control from RAN nodes (primarily, the Central Unit (CU), Distributed Unit (DU), and Radio Unit (RU)) to the RAN Intelligent Controllers (RICs) and the Service Management and Orchestrator (SMO). Specifically, the RICs onboard custom control logic and operate at near-real-time and non-real-time time scales—between 10 ms and 1 s (with custom applications called xApps), or above 1 s (with rApps), respectively—and enforce configurations for the radio resources or deploy policies for system management and optimization. Real-time control (below 10 ms) is provided by dApps, in an extension of the O-RAN architecture. These components are deployed on a heterogeneous set of compute resources, known as the O-Cloud.

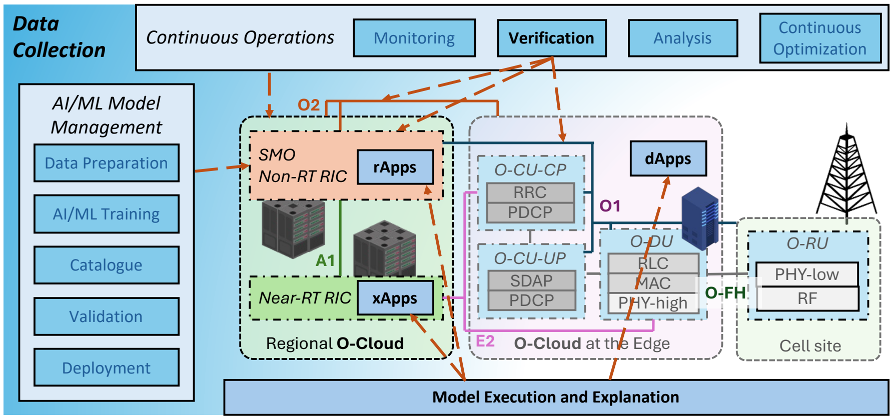

Fig. 1. O-RAN architecture and AI/ML workflows, focusing on model execution, explainability, and verification.

The O-RAN architecture supports several functionalities associated with the lifecycle of AI/ML applications [14], which can be deployed across different endpoints of the architecture. The SMO usually supports data-lake functionalities, collecting telemetry and logs from the thousands of devices and network functions under its purview. It can also host data preparation and model training pipelines, together with a model catalog to deploy trained solutions on the infrastructure (left part of Fig. 1). The Non-RT RIC, often co-located with the SMO, can share some of these responsibilities, together with continuous operations in support of the monitoring and deployment of the models on the infrastructure. Finally, both RICs and the CUs and DUs host model inference, as rApps, xApps, or dApps in the real-time extension of the O-RAN architecture.

When it comes to explainability and verification, rApps, xApps, or dApps often share the need to access to a similar set of telemetry (e.g., Key Performance Indicators (KPIs) from different elements of the RAN, actions and decisions taken by the deployed AI/ML models) and report to a similar set of stakeholders. These include telco personnel, as well as Continuous Integration (CI) and Continuous Deployment (CD) pipelines that manage the models and their deployment. The latter is a critical element in the architecture, as the autonomous and automated nature of O-RAN systems calls for a closed-loop solution for the management of AI/ML models. If verification fails, the model is not apt for controlling or optimizing the network and must be updated, undeployed, or, in more severe cases, replaced entirely or flagged for human operator intervention. The verification results may change dynamically, as the actual network environment and the one known to the model may drift.

There are, however, differences in where verification and explainability algorithms are deployed. Explainability solutions, such as [5], are often coupled with the models, e.g., as extensions in xApps, rApps or components in RICs, to directly interact with the same KPIs data. Verification techniques, instead, track performance trends over time across components, models, and infrastructure, and thus naturally fit into operations and O-Cloud components. Depending on their function, verifiers can be integrated at multiple layers. For instance, the use case in Sec. IV-B relies on analyzing large-scale datasets, multiple model instances, and telemetry. For near-real-time verification of inference outputs, lightweight verifiers like our DT-based solution can be co-located with xApps in the Near-RT RIC, enabling monitoring and feedback at sub-second timescales. Other techniques without such tight timing requirements may access data lakes or infrastructure KPIs to capture overall RAN behavior rather than endpoint decisions. In addition, verification pipelines can also be validated in pre-deployment environments, e.g., digital twins or testbeds.

IV. THE VERIFICATION USE CASE

A. The DRL Agents

We showcase the use of XAI for the verification of the behavior of a relevant AI/ML application for O-RAN systems. Specifically, we consider the case of joint control of RAN slicing and scheduling policies for a set of 5G slices: *(i)* enhanced Mobile Broadband (eMBB); *(ii)* enhanced Machine-type Communications (mMTC); and *(iii)* URLLC. For each slice, the DRL agent [15] selects a RAN slicing policy (i.e., the number of Physical Resource Blocks (PRBs)

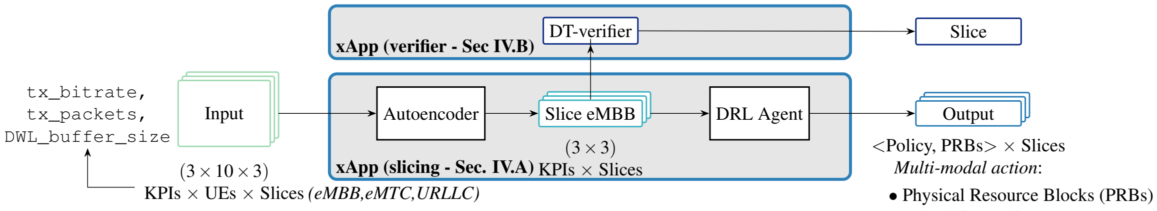

Fig. 2. Operation of the DRL agent xApp and the DT-based verifier xApp

reserved to the slice), and the optimal scheduling policy to serve the User Equipments (UEs) of the slice among Round Robin (RR), Waterfilling (WF), and Proportional Fair (PF). Fig. 2 shows the agent that is implemented as an xApp and provides the appropriate configuration to the next Generation NodeB (gNB) each 250 ms.

Specifically, the bottom of Fig. 2 depicts the architecture of the DRL agent that takes actions to maximize a target reward by monitoring a set of KPIs received from the gNB via the E2 interface and processed by an autoencoder. The KPIs are transmission bitrate in Mbps, number of transmitted packets, and size of the downlink (DWL) buffer in bytes. These metrics are collected as individual measurements for each user of every slice.

In this work, we leverage a public dataset [5] that analyzes two different DRL agents: the eMBB-oriented agent strives to maximize the transmission bitrate, while the URLLC-oriented agent minimizes DWL buffer occupancy, which acts as a proxy for latency. The dataset contains per-user information on the above-mentioned KPIs, among others, as well as logs of the agents' operation (observed state, action and reward) in an urban scenario emulated with the Colosseum Open RAN digital twin. We refer the reader to [5] for the complete details of the dataset. Since the dataset provides a low number of samples for the eMBB-oriented agent, we augmented the samples to be 10 000 via Conditional Tabular GAN (CTGAN), a generative model tailored for tabular data that preserve the joint KPI–slice distributions.

B. Verifying Agent Behavior with Decision Trees

For our verification use-case, we address the following question:

> *Is it possible to infer which is the slice a user belongs to and whether the agent decision is adequate for that slice by only observing the input KPIs (to the model) of a user?*

The question may appear obvious as the mobile network operator has full knowledge about the user-to-slice mapping. However, this simple question makes it possible to verify whether the agent operation is consistent with the knowledge it acquired during training. Since the agent was trained to perform joint RAN slicing and scheduling control over the three slices simultaneously, each slice has a unique fingerprint in terms of resource assignment. The capability of verifying the agent's behavior is instrumental to troubleshoot improper or conflicting configurations generated by different xApps with diverse reward functions and target KPIs.

For verification, we propose to use DTs, a popular method commonly used for classification tasks. DTs classify a population into branch-like segments that construct an inverted tree with a root node, internal nodes, and leaf nodes. Being a non-parametric algorithm, DTs efficiently manage large datasets without imposing a parametric structure, handle well both categorical and numerical data, and are robust to outliers and missing data which is a common case in RAN telemetry. DTs align well with RAN constraints: they are fast to train and infer, and naturally interpretable. Unlike attribution methods such as SHAP or LRP, which often incur significant computational overhead, DTs enable slice verification at near-real-time timescales, making them deployable alongside xApps in the Near-RT RIC. Compared to neural-symbolic approaches or probabilistic verification, DTs offer a pragmatic balance between interpretability and operational feasibility at stringent timescales.

The top of Fig. 2 and Fig. 3 illustrate how the slice-verifier is integrated within the normal operation of the DRL-based xApp, and showcases its capabilities in a few examples. Specifically, the slice-verifier can identify and signal *a)* drifts in feature space for a specific slice, *b)* misclassifications based on trained data can identify bias or error in learning, and *c)* conflicting predictions for similar feature space that might be caused due to approximations

TABLE I
METRICS OF SLICE-VERIFIER FOR EMBB-ORIENTED AGENT

SLICE	PRECISION	RECALL	F1-SCORE	SUPPORT
eMBB	0.81	0.81	0.81	3297
mMTC	0.82	0.82	0.82	3479
URLLC	0.81	0.81	0.81	3224
Accuracy		0.82		10000
Macro Avg.	0.82	0.82	0.82	10000
Weighted Avg.	0.82	0.82	0.82	10000

TABLE II
METRICS OF SLICE-VERIFIER FOR URLLC-ORIENTED AGENT

SLICE	PRECISION	RECALL	F1-SCORE	SUPPORT
eMBB	0.99	0.98	0.99	3392
mMTC	0.86	0.87	0.87	3280
URLLC	0.87	0.87	0.87	3328
Accuracy		0.91		10000
Macro Avg.	0.91	0.91	0.91	10000
Weighted Avg.	0.91	0.91	0.91	10000

or coarse feature transformations. We train the slice-verifier with XGBoost using as input features those of the agent and as output classes the labels of the slices. We tune XGBoost and DT hyperparameters to balance expressiveness (by capturing non-linear relations in KPIs and ensuring all features contribute per splits) and lightweight inference (by restricting the maximum depth and minimum samples per split).

To demonstrate that the verification could be effective and lightweight, we restrain our support to train the slice-verifier to a small subset (less than 0.1%) of samples that were used to train the agent. We test the accuracy using using standard classification metrics like “Precision”, “Recall” and “F1-score”. Tables I and Table II show the accuracy of the DT slice-verifier for the eMBB- and URLLC-oriented agents in all slices. We observe that the agents' operation is robust because the slice-verifier achieves high accuracy in all the metrics, meaning that it is capable to correctly identify samples and avoid false positives. Further, across all the experiments that were carried out with a varying number of users, the variance of the input KPI is low. The accuracy is higher for the URLLC-oriented agent as its reward function yields KPI distributions with clear patterns (low buffer). In contrast, the eMBB-oriented agent faces more trade-offs to optimize throughput, yielding KPI distributions with some overlap with the mMTC slice, which makes the verification harder and explains the lower accuracy values.

While DTs offer lightweight interpretability, they also come with limitations. They may oversimplify complex, high-dimensional feature relationships that advanced models, instead, capture. Simplicity is a strength in our case, but may not generalize to all AI/ML applications within Open RAN. Moreover, DT-based verification focuses on consistency in behavior, rather than formal guarantees. Future work should investigate hybrid methods, e.g., combining DTs with formal verification or counterfactual reasoning, to enhance trustworthiness.

V. OPEN CHALLENGES AND FUTURE DIRECTIONS

As the need for network automation within the Open RAN grows, so does the importance of ensuring the trustworthiness of AI/ML operations. This section explores fundamental challenges ahead for AI verification and XAI, including tailoring the verification to Open RAN tasks, scalability of the proposed solutions, and prototyping and testing.

a) Open RAN-Oriented Verification: The current approach for AI verification, which focuses on individual models in isolation, is insufficient for the RAN ecosystem's numerous interacting components [7], [8], [9]. Therefore, it is necessary to develop verification techniques that extend beyond a single AI/ML model to provide system-level guarantees. Standardizing the integration of explainability and robustness checks into frameworks like ZSM is an important research direction to enable trustworthy automated resource management [12].

b) Tradeoff Scalability-Soundness: The computation time for highly sound verification techniques is very high, making them impractical for real-time operations. A promising way to overcome this challenge is to define system-specific boundaries to reduce the solving space and use explainability to understand the limits of the model's operation.

c) Prototyping and Testing: Prototyping and testing verification and XAI solutions requires a phased approach, involving multiple resources from testbeds used as sandbox for development, to production network environments that can offer access to data at scale. This is necessary to test the scalability of proposed AI verification techniques, and of the control and optimization architectures enabling them.

VI. CONCLUDING REMARKS

This paper analyzed XAI and AI verification techniques and outlined their applicability to Open

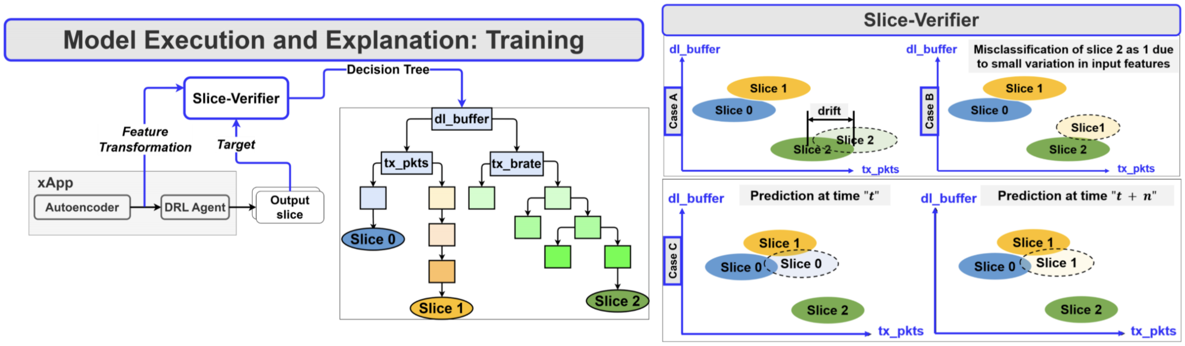

Fig. 3. The slice-verifier: training and verification capabilities. Case A shows a "drift" in the feature space for a specific slice, case B highlights a misclassification, and case C illustrates conflicting predictions for data points in a similar feature space at different times.

RAN. The unique properties of the RAN ecosystem call for tailored verification approaches that can operate across multi-vendor systems and at timescales compatible with RAN control. Through our use case, we showed that DRL decisions for slicing and scheduling can be effectively verified using lightweight DT-based models, ensuring consistency with training knowledge. While XAI and AI verification in Open RAN are still in their early stages, our study provides a foundation for advancing trustworthy AI/ML adoption in this domain.

REFERENCES

[1] M. Polese *et al.*, "Understanding O-RAN: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1376–1411, 2023.

[2] "RAN automation, SON, RIC, xApps & rApps in the 5G era: 2024 – 2030 – opportunities, challenges, strategies & forecasts," https://www.snstelecom.com/son, [Accessed 21-10-2025].

[3] N. C. Luong *et al.*, "Applications of deep reinforcement learning in communications and networking: A survey," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3133–3174, 2019.

[4] N. M. Yungaicela-Naula *et al.*, "Misconfiguration in O-RAN: Analysis of the impact of AI/ML," *Computer Networks*, vol. 247, p. 110455, 2024.

[5] C. Fiandrino *et al.*, "Explora: AI/ML explainability for the open RAN," *Proc. of the ACM on Networking*, vol. 1, no. CoNEXT3, pp. 1–26, 2023.

[6] J. M. Wing, "Trustworthy AI," *Commun. ACM*, vol. 64, no. 10, p. 64–71, 2021.

[7] J. Liu *et al.*, "Abstraction and refinement: Towards scalable and exact verification of neural networks," *ACM Trans. Softw. Eng. Methodol.*, vol. 33, no. 5, Jun 2024.

[8] C. Urban *et al.*, "A review of formal methods applied to machine learning," *arXiv preprint arXiv:2104.02466*, 2021.

[9] L. Weng *et al.*, "PROVEN: Verifying robustness of neural networks with a probabilistic approach," in *Proc. of International Conference on Machine Learning*, 2019, pp. 6727–6736.

[10] T. Senevirathna *et al.*, "A survey on XAI for 5G and beyond security: Technical aspects, challenges and research directions," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2024.

[11] J. Marques-Silva *et al.*, "Delivering trustworthy AI through formal XAI," *Proc. of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 11, pp. 12 342–12 350, Jun. 2022.

[12] N. Khan *et al.*, "Explainable and robust artificial intelligence for trustworthy resource management in 6G networks," *IEEE Communications Magazine*, vol. 62, no. 4, pp. 50–56, 2024.

[13] M. Landers *et al.*, "Deep reinforcement learning verification: A survey," *ACM Comput. Surv.*, vol. 55, no. 14s, jul 2023.

[14] O-RAN Working Group 2, "O-RAN AI/ML Workflow Description and Requirements 1.03," O-RAN.WG2.AIML-v01.03 Technical Specification, July 2021, [Accessed 21-10-2025].

[15] M. Polese *et al.*, "ColO-RAN: Developing machine learning-based xApps for Open RAN closed-loop control on programmable experimental platforms," *IEEE Transactions on Mobile Computing*, pp. 1–14, 2022.

BIOGRAPHIES

Rahul Soundrarajan is a Principal Engineer at Tejas Networks. Prior to this, he was an entrepreneur & independent consultant. He has designed, developed & evaluated ML Algorithms for Near-RT RIC, Non-RT RIC and network optimization use cases for which he holds several patents. Contributions to this work was done by Rahul as an independent research consultant prior to joining Tejas Networks.

Claudio Fiandrino is a Research Assistant Professor at IMDEA Networks Institute, Spain, where he leads the Laboratory for Resilient AI Networking. His expertise and interests lie at the interface of explainable and robust AI/ML and mobile networks.

Michele Polese is a Research Assistant Professor at the Institute for the Wireless Internet of Things at Northeastern University, Boston. He received his Ph.D. in Information Engineering at the University of Padova in 2020. He then joined Northeastern University as a research scientist. His research interests are in the analysis and development of protocols and architectures for future generations of cellular networks.

Salvatore D'Oro is a Research Associate Professor with Northeastern University. He received his Ph.D. from the University of Catania in 2015. His research focuses on optimization and learning for NextG systems and the Open RAN.

Leonardo Bonati is an Associate Research Scientist at the Institute for the Wireless Internet of Things, Northeastern University, Boston. He received a Ph.D. degree in Computer Engineering from Northeastern University in 2022. His research focuses on softwarized approaches for the Open RAN of next generation of cellular networks.

Tommaso Melodia received a Ph.D. in Electrical and Computer Engineering from the Georgia Institute of Technology in 2007. He is the William Lincoln Smith Professor at Northeastern University, the Director of the Institute for the Wireless Internet of Things, and the Director of Research for the PAWR Project Office. His research focuses on wireless networked systems.