Closed-loop Intelligence Using Large Language Models in Wireless Networks

Hojjat Navidan*, Mohamed Seif†, H. Vincent Poor†, Ingrid Moerman*, Adnan Shahid*

*IDLab, Department of Information Technology, Ghent University – imec, Belgium
†Department of Electrical and Computer Engineering, Princeton University, USA
Email: {hojjat.navidan, adnan.shahid, ingrid.moerman}@ugent.be, {mseif, poor}@princeton.edu

***Abstract*—Large Language Models (LLMs) are emerging as a transformative tool for wireless network management, offering the ability to interpret natural language instructions, reason over diverse data types, and adapt rapidly through in-context learning. Their generalization capabilities make them promising candidates for acting as the intelligent core within the Open Radio Access Networks (O-RAN), enabling more robust and flexible control of network behavior. However, it remains unclear how to systematically integrate, evaluate, and compare different LLMs for specific tasks in live network deployments. We present Closed-loop Intelligence using LLMs (CLI-LLM), an open-source framework that transforms real-time network metrics and operator intents into automated network control actions. The goal of this demo is to showcase how different LLMs, whether commercial or open-source, can be integrated into the RAN Intelligent Controller (RIC) to handle tasks such as resource allocation, while also providing time-series analysis capabilities, including anomaly detection and trend prediction.**

***Index Terms*—Large Language Models, Network Management and Orchestration, O-RAN**

I. Introduction

The Open Radio Access Network (O-RAN) paradigm introduces programmability into cellular networks via the RAN Intelligent Controller (RIC) [1]. While the RIC enables intelligent control through applications (xApps), programming them for every complex scenario remains a challenge. Large Language Models (LLMs) present a transformative solution, offering the ability to reason, interpret high-level natural language goals, and adapt without task-specific retraining [2]. However, there is a gap between conceptual proposals and practical, closed-loop implementations that connect live network data to LLM-driven control. While prior studies have shown that LLMs can parse intent or optimize resource allocation through handcrafted prompts, these systems typically rely on static inputs and remain open-loop.

In this demo, we present CLI-LLM[1], an open-source framework that addresses this gap by enabling autonomous, closed-loop intelligence using LLMs in wireless networks. The system ingests real-time Key Performance Metrics (KPMs) and operator-defined Service Level Agreements (SLAs) as instructions and generates control actions via an LLM. Crucially, unlike conventional ML models, CLI-LLM leverages the LLM's zero-shot generalization, allowing operators to define novel and complex control objectives in natural language without any model retraining.

While the framework is designed to be modular and generic, making it applicable to a wide range of network environments, this demo specifically showcases its integration with an O-RAN system. CLI-LLM can be used both for data-driven analytics, such as anomaly detection, traffic prediction, and diagnostics, as well as in closed-loop scenarios where direct control over the network is possible. Furthermore, it lays the groundwork for future research on systematically analyzing and comparing how different LLMs reason under various prompting strategies for diverse network management tasks.

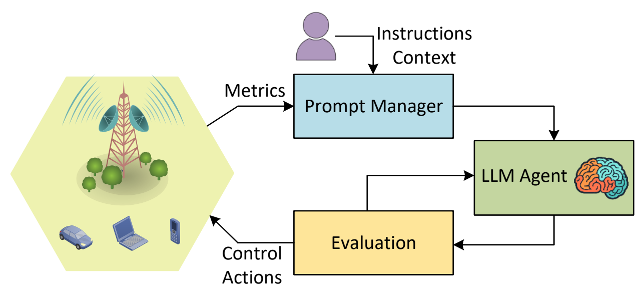

Fig. 1. Conceptual architecture of the CLI-LLM.

II. System Design

The proposed framework introduces a closed-loop control system that integrates real-time network telemetry, high-level instructions, and LLM-based reasoning. As illustrated in Figure 1, the architecture consists of three core components: the prompt manager, the LLM agent, and the evaluation module. The prompt manager transforms raw KPMs, SLA objectives, and contextual inputs into structured and meaningful prompts, serving as the interface between the live network and the LLM.

These prompts are formatted in a machine-readable structure such as JSON and include not only the KPM values but also semantic descriptions, units, and relevant metadata to further assist the LLM. They are then passed to the LLM

This work was supported by the Horizon Europe program under the MCSA Staff Exchanges 2021 grant agreement 101086218 (EVOLVE project) and the Horizon-JU-SNS-2023 Research and Innovation Program under Grant Agreement No. 101139194 (6G-XCEL project).

[1]https://github.com/hnavidan/CLI-LLM

© 2025 IFIP

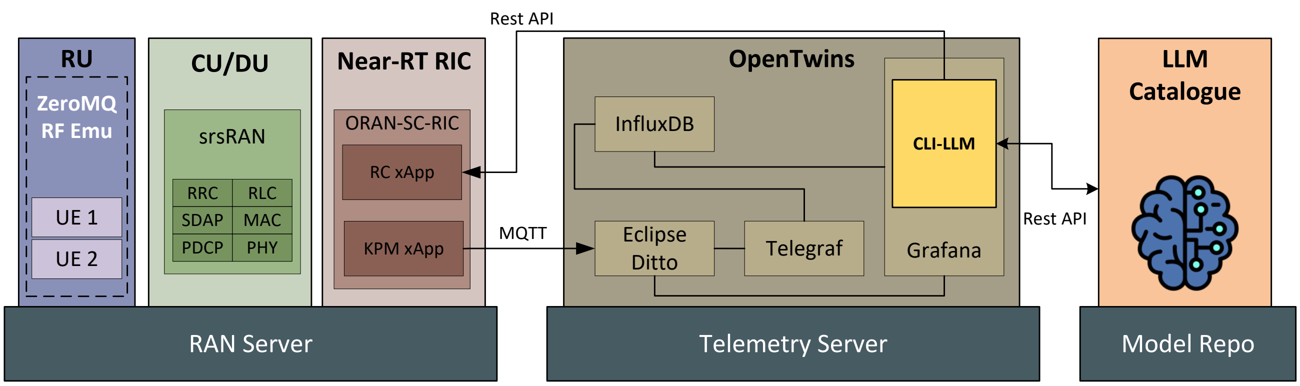

Fig. 2. Deployment architecture of the demonstration.

agent, which generates high-level control decisions—such as adjusting resource allocation policies—based on both current network conditions and historical context. Finally, the evaluation module validates these decisions against predefined policies and safety constraints before they are applied as control commands in the network.

This conceptual model is realized by the detailed implementation architecture shown in Figure 2. The system comprises three domains: a live O-RAN environment, a data collection and storage platform, and a decoupled LLM intelligence layer. The O-RAN environment is built using the srsRAN stack [3] and the O-RAN Software Community (ORAN-SC) RIC [4]. A KPM xApp streams real-time metrics to a telemetry server based on the open-source OpenTwins platform [5], where the network state is visualized and stored.

The prompt manager is embedded within Grafana on the telemetry server, enabling operators to construct context-rich prompts by combining real-time network state with instructions. For instance, instructions can be abstract policy objectives, such as "prioritize latency for slice A," instead of fixed rules. These prompts are sent to the standalone LLM agent, which can either run locally on the infrastructure or interface with a commercial LLM service.

The LLM processes the input and returns its control recommendation. Once evaluated, this control command is forwarded to the RAN Control (RC) xApp, which enforces the decision within the live network. This end-to-end framework demonstrates how LLMs can be integrated within the O-RAN architecture to drive intelligent, automated control through open-source and modular tools, thereby completing the closed-loop operation of the system.

III. Demo Setup

The demo showcases dynamic resource allocation between two network slices using an entirely open-source testbed. Participants will interact with the system through a Grafana-based control panel to issue commands and observe the closed-loop process. To demonstrate the LLM's reasoning capabilities, attendees can input complex, conditional instructions that are difficult to program with static rules. Figure 3 illustrates this end-to-end flow, from a natural language prompt to a validated control action.

Furthermore, the demo highlights the framework's modularity by enabling attendees to switch between different LLM backends, such as GPT-4 and Gemini 2.5, and directly observe how their control decisions differ for the exact same natural language command. This flexibility extends to interfacing with both commercial API-based services and locally hosted open-source models, accommodating diverse operational requirements and privacy constraints. In addition to control, the system also enables the use of the LLM for exploratory analysis of KPM data.

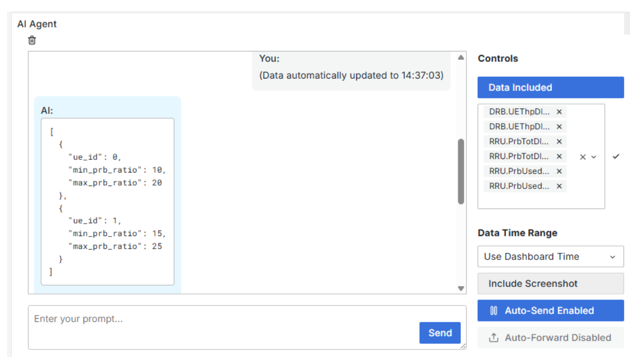

Fig. 3. An example of the resource allocation decision-making process within the Grafana-based LLM control panel.

References

[1] J. F. Santos, A. Huff, D. Campos, K. V. Cardoso, C. B. Both, and L. A. DaSilva, "Managing O-RAN Networks: xApp Development From Zero to Hero," *IEEE Communications Surveys & Tutorials*, pp. 1–1, 2025.

[2] P. Gajjar and V. K. Shah, "Oransight-2.0: Foundational llms for o-ran," *IEEE Transactions on Machine Learning in Communications and Networking*, vol. 3, pp. 903–920, 2025.

[3] S. R. Systems. (2025) srsRAN Project: Open source O-RAN 5G CU/DU solution. [Online]. Available: https://github.com/srsran/srsRAN_Project

[4] O.-R. S. Community. (2025) O-RAN SC Projects — ran intelligent controller. [Online]. Available: https://docs.o-ran-sc.org/en/latest/projects.html

[5] J. Robles, C. Martín, and M. Díaz, "OpenTwins: An open-source framework for the development of next-gen compositional digital twins," *Computers in Industry*, vol. 152, p. 104007, Nov. 2023.