In: IEEE 102nd Vehicular Technology Conference (VTC2025-Fall), 19-22 Oct 2025, Chengdu, China
DOI: 10.1109/vtc2025-fall65116.2025.11310512

End-to-End Edge AI Service Provisioning Framework in 6G ORAN

Yun Tang
yun.tang@cranfield.ac.uk

Udhaya Chandhar Srinivasan
u.srinivasan@cranfield.ac.uk

Benjamin James Scott
benjamin.scott@cranfield.ac.uk

Obumneme Umealor
obumneme.umealor@cranfield.ac.uk

Dennis Kevogo
Dennis.Kevogo@cranfield.ac.uk

Weisi Guo
weisi.guo@cranfield.ac.uk

***Abstract*—As 6G networks evolve to support pervasive AI-driven applications, seamless provisioning of Edge AI services has become increasingly vital. However, current orchestration processes remain fragmented, requiring extensive coordination between AI-powered application developers and the network operators. In this paper, we propose a novel end-to-end orchestration framework that integrates Large Language Model (LLM) agents into O-RAN to automate edge AI service subscription and deployment. Our system translates high-level user intents into orchestrated workflows, including AI model selection, mobility-aware placement, and performance monitoring. We demonstrate the framework via a prototype built on our open-source O-RAN simulator, showcasing intelligent, intent-driven AI service provisioning. This work represents a key step toward AI-native, accessible, and scalable service management in 6G.**

***Index Terms*—Edge AI-as-a-Service, 6G, O-RAN, LLM Agent**

I. Introduction

Edge intelligence is becoming increasingly important as next-generation use cases demand low-latency AI services close to end users (as shown in Fig. 1). However, deploying AI models at the network edge and configuring the underlying infrastructure remains complex and time-consuming. Without an automated solution, use case developers must manually integrate AI applications with distributed edge resources and confer with network service providers to tailor network settings for each use case, distracting from their primary goal of building innovative applications. Thus, there is a growing need for an end-to-end orchestration framework that abstracts these low-level tasks, allowing use case innovators to focus on application logic rather than the intricacies of AI service deployment and network configuration.

In parallel, the evolution toward 6G networks is expected to embrace AI deeply into the network, for both network operations and the connected use cases [1]. Research visions for 6G emphasize AI-native network management, where AI functions are integrated across cloud, core, and radio domains. For instance, the O-RAN architecture [2] already introduces RAN Intelligent Controllers (RICs) to enable data-driven control in the radio access network. Yet, current frameworks lack an end-to-end mechanism for orchestrating use case-facing AI services across domains in a holistic manner.

Recent advances in large language models (LLMs) [3], [4] open up new possibilities: LLMs can interpret high-level intents, reason about complex tasks, and even generate configuration or code, which can be leveraged to automate network and service management.

Motivated by these trends, we propose an LLM-powered, end-to-end orchestration framework to simplify Edge AI service deployment in 6G O-RAN networks. Our framework enables developers to describe their application needs using natural language, with an intelligent agent translating these intents into mobility-aware AI service provisioning actions.

The key contributions of this paper are:

- We design an intent-driven orchestration framework that integrates LLM agents with O-RAN architecture to automate Edge AI service provisioning.
- We introduce a subscription-based model to manage AI service lifecycle, enabling mobility-aware deployment and dynamic reconfiguration.
- We demonstrate the proposed framework through an open-source O-RAN simulator, showcasing end-to-end automation from user interaction to service deployment.

The following sections detail the architecture and components of our framework, describe its implementation, and present a working demonstration.

II. Related Works

A. Edge AI Orchestration Approaches

Prior works have explored platforms to orchestrate compute and AI workloads at the network edge. For example, Oakestra [5] is a lightweight hierarchical orchestrator for edge computing that tackles challenges like unreliable links and diverse hardware by federating resources and delegating tasks efficiently. Existing edge orchestration solutions primarily focus on container management, workload placement, and latency optimization. However, they often do not specifically address AI service user engagement. Our work differs by leveraging the LLM-based agents to interact with the users to streamline their experience in subscribing to edge AI services.

The authors are with the School of Aerospace, Transport and Manufacturing (SATM), Cranfield University, United Kingdom. The work is supported by EPSRC CHEDDAR: Communications Hub for Empowering Distributed clouD computing Applications and Research (EP/X040518/1) (EP/Y037421/1).

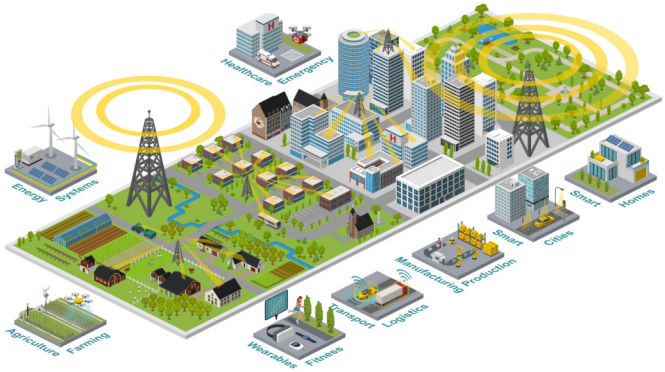

Fig. 1. Edge AI use cases supported by 6G networks, emphasizing the need for low-latency edge intelligence.

B. O-RAN and 6G

The O-RAN initiative [2] specifies a disaggregated RAN architecture with open interfaces and intelligent controllers to enable innovation in network management. In O-RAN, a near-real-time RAN Intelligent Controller (near-RT RIC) hosts xApps that perform closed-loop control of the RAN (e.g., scheduling, slicing) on the order of 10 ms to 1 s, while a non-real-time RIC (within the Service Management and Orchestration framework) handles policy and analytics on > 1s timescales. The RIC concept is a cornerstone that brings AI and automation into RAN operations. Early O-RAN deployments and research (often on 5G testbeds) have shown the feasibility of dynamic RAN optimization via xApps [6]. Looking ahead, AI in 6G will not only optimize the RAN but also coordinate end-to-end network behaviour across domains (i.e., RAN, edge, core, and cloud) [7]. This work builds on this background by using the O-RAN platform (RIC applications in RAN) as the vehicle for automated AI service subscription and deployment management.

C. LLMs in 6G Networks

The emergence of large language models has prompted exploration into their role within future network management and orchestration. For example, Maestro [8] proposes a framework where multiple LLM-based agents (representing multiple stakeholders such as network service providers) negotiate shared network resources. More generally, it is highlighted that LLMs can grasp user intent, reason about tasks, and execute commands, potentially redefining how we interact with and control network services [9]. Our proposed framework can be seen as part of this emerging paradigm: it uses an LLM-based agent as an intelligent orchestrator that bridges the gap between a use case's high-level intent and low-level service provisioning actions.

III. The Framework

Fig 2 presents an overview of the proposed framework. At the core of the framework is an LLM-based orchestrator agent that serves as the "brain" of the system. When a user describes a use case (for example, "I'm building a robot dog fleet to search for stray animals in ..."), the request is forwarded to the LLM agent. The agent is equipped with tools to interact with the end user, retrieve necessary knowledge (AI service details, user AI service subscription history, new use case demands, and user equipment information), and manage AI service subscriptions. The LLM's ability to perform reasoning over the user's use case needs is what differentiates this framework from static rule-based orchestrators. The Non-RT RIC manages the AI service subscriptions and deployments depending on the connectivity and mobility of the subscribing user equipments within the network.

A. AI Service Recommendation and Selection

One of the first tasks is to search and select a suitable AI service that fulfils the use case's needs. The framework includes an AI service registry that catalogues available pre-trained and edge-deployable models (and possibly training pipelines) for various AI tasks (see [10]). Upon analysing the use case scenario, the LLM agent queries this repository to find AI services that fulfil the functionality requirements. For instance, if the request is for an object detection service, the repository might have services serving a trained ResNet or YOLO detector. The agent can consider factors such as the model's accuracy (if the AI model is benchmarked against the user's scenario), input requirements (camera feed, sensor data, etc.), and resource footprint when recommending the services. The output of this step is a list of AI service candidates, and the user is then prompted to make a selection. This task aims to abstract the complexity of AI model selection: instead of the user having to search, choose, and configure a model, the orchestrator's intelligence handles it. In a more advanced implementation, this step could also involve model optimization – for example, choosing a quantized model for faster inference if accuracy is not the developer's primary concern, or an explainable model for ethical and legal compliance.

B. AI Service Subscription

In line with mobile network paradigms, we propose a subscription-based mechanism for AI service provisioning. Once the user selects an AI service—either manually or with agent assistance—the agent will initiate the subscription workflow, where the agent requests user equipment (UE) identifiers, such as device IDs or IMSI-equivalent tokens, which are used to create the AI service subscriptions.

AI service subscriptions, which encapsulate a selected AI service and the subscribing UEs, are stored in the RIC's subscription database. This process is facilitated by the *AI Service Subscription Management* rApp, which exposes a full CRUD (Create, Retrieve, Update, Delete) API. This API allows the agent to dynamically update or revoke subscriptions based on changes in user needs and service availabilities. In addition, the subscription database also enables retrieval-augmented generation (RAG) during the user intent profiling stage, where the agent can perform personalized recommendations according to the user's historical subscriptions.

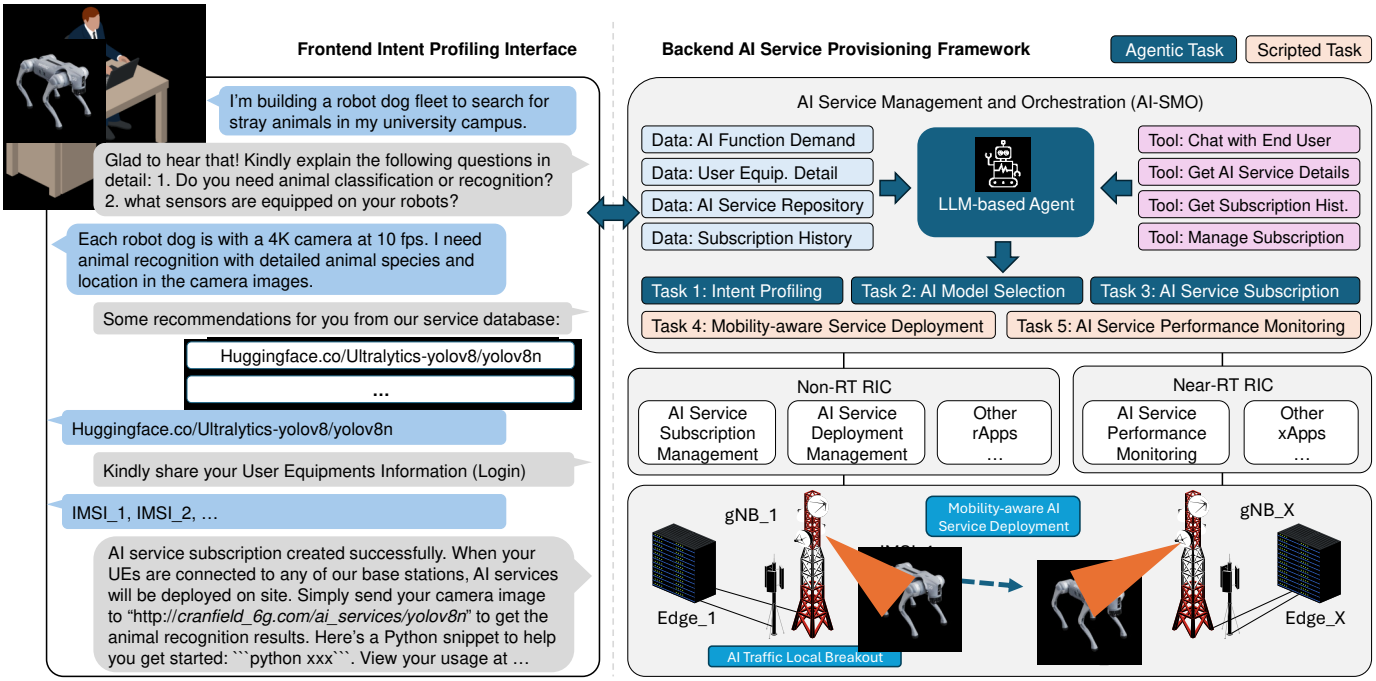

Fig. 2. End-to-End Edge AI Service Provisioning Framework Overview.

The subscription-based mechanism provides a streamlined and intuitive experience not only for end users but also for the agentic orchestrator, which we believe is at least as important as the user experience, if not more so. Furthermore, this mechanism enables flexible future extensions in commercial deployment, such as premium subscription tiers with guaranteed QoS (e.g., latency and bitrate constraints), auto-scaling policies, explainability interfaces for regulatory compliance, and fallback deployment strategies for safety-critical use cases.

C. Mobility-aware AI Service Deployment

To translate static AI service subscriptions into mobility-aware AI service deployments across the network, the framework includes a specialized rApp called the *AI Service Deployment Manager*. Its role is to dynamically deploy or undeploy AI services at the base stations (specifically, their co-located edge servers) in response to AI service subscription, user equipment (UE) mobility, and connectivity changes.

For each active AI service subscription, the deployment manager continuously scans all base stations to detect whether any of the subscribing UEs are currently connected. If a base station has at least one subscribing UE, the corresponding AI service is either maintained (with its undeployment countdown reset) or newly deployed. Deployment is contingent upon at least two checks: compatibility between the AI service and the edge infrastructure (e.g., software and hardware compatibility), and the availability of sufficient CPU and accelerator device memory resources. The deployment manager interacts with an edge infrastructure manager (e.g., a Kubernetes cluster at the edge, an ETSI MEC platform, or a virtualization layer) to launch the AI service, which allocates necessary resources (vCPUs, memory, and GPU) to the AI service container. Networking between the RAN and the AI service is then set up via local breakout rules, ensuring the service is reachable from the RAN (to ingest data or serve results). To use the AI service, UEs can simply post requests to a common URL such as "*https://cranfield-6g.com/ai_service/microsoft_resnet_50*" and the base station will forward the request to the edge server if available.

If no subscribing UEs are connected to a base station currently hosting a deployment, a countdown mechanism is triggered. This delay avoids rapid redeployment cycles in response to transient disconnects (e.g., caused by ping-pong effects). Once the countdown reaches zero, the AI service is undeployed, and associated configurations (e.g., local breakout rules, QoS monitoring) are cleaned up.

This deployment logic ensures that AI services are autonomously provisioned only where they are needed, optimizing resource utilization while adapting to UE mobility in real time. The core steps of this process are summarized in Algorithm 1.

D. Service Monitoring

To ensure that the deployed AI service meets the expected performance, our framework includes a Service Monitoring xApp to monitor AI service traffic and alert on QoS-related issues, presenting both real-time and historical insights to the user.

QoS Monitoring Depending on the available service models or API endpoints, the key QoS metrics related to the AI service traffic can include:

Algorithm 1: Mobility-Aware AI Service Deployment

Input: AI service subscriptions from the database

```
foreach subscription S do
    foreach base station B do
        if any UE in S is served by B then
            foundUE ← True;
        else
            foundUE ← False;
        if AI service is already deployed on the edge server at B then
            deployed ← True;
        else
            deployed ← False;
        if foundUE then
            if deployed then
                Reset undeployment countdown;
            else
                if AI service is compatible with edge server at B then
                    if CPU and accelerator device memory is sufficient then
                        Deploy AI service at B;
                        Update deployment data at B;
                        Set up local breakout rule at B;
                        Start AI service performance monitoring xApp;
                    else
                        Report: insufficient resources for deployment;
                else
                    Report: AI service not compatible with this node;
        else
            if deployed then
                Decrement countdown;
                if countdown is 0 then
                    Undeploy AI service container;
                    Clean breakout and monitoring rules;
```

- **Latency**: End-to-end delay from the AI service request to response.
- **Throughput**: The bandwidth allocated and utilized for AI service communication.
- **Packet Loss**: Identifying network congestion that could degrade AI service quality.
- **Jitter**: Variability in packet transmission times that might affect real-time AI applications.
- **Service Availability**: Detecting disruptions in AI model inference or edge server downtime.

Reporting and Adaptations The monitoring xApps generate reports for the user, which can include:

- **Real-time QoS Dashboards** Graphical reports summarizing AI service performance and network conditions.
- **Proactive Notifications** Alerts when QoS degradation is predicted, allowing users to take preventive action.
- **Network Adaptations** Based on trends, the deployment manager may recommend actions such as requesting a prioritized slice, scaling the AI service deployments up or down, or providing explanations to manage users' expectations or adjust use case requirements.

Such a closed-loop feedback mechanism can keep users informed about the status of their requested AI service, offering transparency and fostering greater user trust.

IV. Demonstration

A. Prototype Implementation

Considering the diverse choices of RAN, RIC, core, and edge solutions in the market with different interfaces and functionality architectures, we implement the proposed AI service provisioning framework on top of our open-sourced O-RAN experiment platform and conduct an end-to-end demonstration to showcase the feasibility of the envisioned framework (see [11]). Specifically, the platform consists of a web-based frontend and a backend network simulator. The frontend visualizes the simulated network, presents network live data dashboards, and most importantly, the end user chat interfaces with the AI service orchestrator agent. The (lightweight python-only) backend consists of three functional layers: the network simulation layer which simulates the dynamics of UEs, base stations, cells, edge, core, and RIC, the network knowledge layer which serves agent-friendly network knowledge (including the AI service registry) by digesting live network data from the underlying network layer, and lastly, the intelligence layer hosts the LLM-powered agents (e.g., AI service intent profiling agent and AI service deployment agent) to drive the AI service provisioning pipelines. The LLM agent in our prototype is implemented using OpenAI Agent SDK [12] and powered by GPT-4.1 (through API) [13]. The AI services are compiled based on pre-trained models from HuggingFace [14].

B. Demonstration

Fig. 2 presents an actual conversation where a simulated use case developer chats with the AI service orchestration

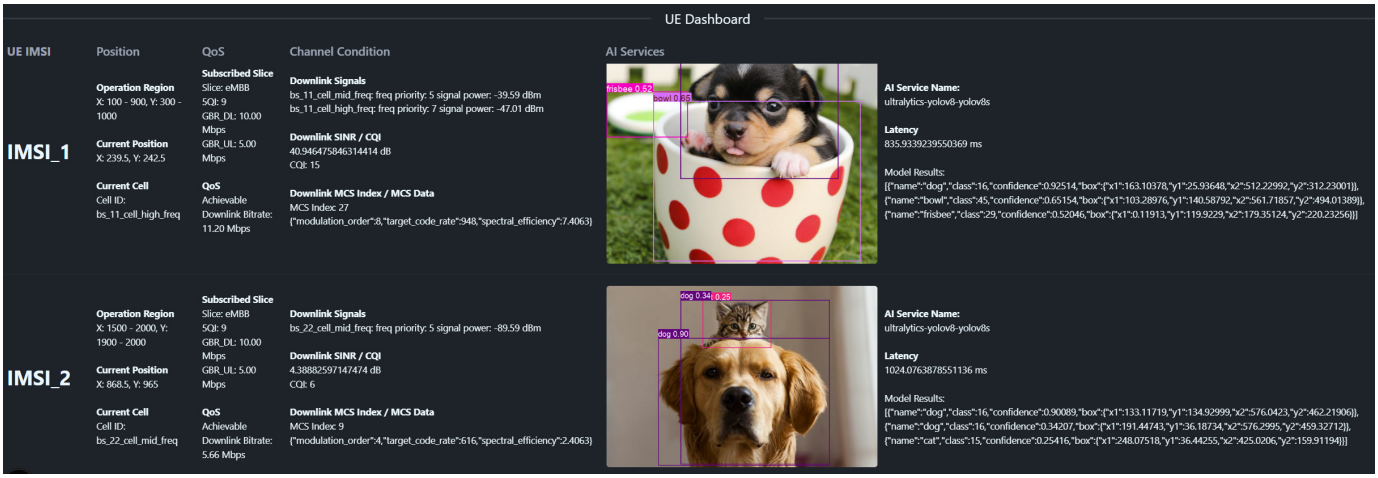

Fig. 3. UE Dashboard screenshot of the platform's frontend interface, displaying the latest AI service request responses of the two subscribing UEs.

agents to subscribe to an object (animal) recognition AI service. It shows that the agent can 1) explore the AI service database and recommend appropriate AI services, 2) ask for necessary information such as UE IDs and create an AI service subscription, and 3) explain how the AI services are provisioned and used, including the API calls and code snippets. Fig. 3 is the screenshot of the UE Dashboard of the frontend interface, displaying the latest AI service response. Readers are welcome to try out our framework at [11].

V. Conclusion

This paper introduced an end-to-end Edge AI service provisioning framework that integrates Large Language Model (LLM) agents into the 6G O-RAN architecture. The proposed framework automates the complex process of deploying AI services at the network edge by translating high-level user intents into actionable tasks such as AI model selection, service subscription, mobility-aware deployment, and QoS monitoring. Through an open-source prototype built atop a custom O-RAN simulator, we demonstrated the viability of this approach, showcasing seamless orchestration from user interaction to AI service automation.

Our demonstrated framework highlights the potential of 1) opening up new revenue streams by offering edge AI service subscriptions, 2) accelerating the development of innovative AI-driven use cases, 3) enhancing user experiences, and 3) reducing user engagement cost. Future work will explore the real-world integration with commercial RAN platforms, quality-of-AI-service guarantees, multi-vendor compatibility, and continuous learning capabilities for improved orchestration in diverse, large-scale environments.

Acknowledgment

AI tools (ChatGPT, DeepSeek) have been used to revise (grammar and organization) author-written content.

References

[1] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6g: Ai empowered wireless networks," *IEEE communications magazine*, vol. 57, no. 8, pp. 84–90, 2019.

[2] O-RAN Alliance, "O-ran alliance e.v," https://www.o-ran.org/, 2023, accessed: 2025-03-12. [Online]. Available: https://www.o-ran.org/

[3] H. Zhou, C. Hu, Y. Yuan, Y. Cui, Y. Jin, C. Chen, H. Wu, D. Yuan, L. Jiang, D. Wu *et al.*, "Large language model (llm) for telecommunications: A comprehensive survey on principles, key techniques, and opportunities," *IEEE Communications Surveys & Tutorials*, 2024.

[4] S. Long, F. Tang, Y. Li, T. Tan, Z. Jin, M. Zhao, and N. Kato, "6g comprehensive intelligence: network operations and optimization based on large language models," *IEEE Network*, 2024.

[5] G. Bartolomeo, M. Yosofie, S. Bäurle, O. Haluszczynski, N. Mohan, and J. Ott, "Oakestra: A lightweight hierarchical orchestration framework for edge computing," in *2023 USENIX Annual Technical Conference (USENIX ATC 23)*, 2023, pp. 215–231.

[6] X. Limani, A. Troch, C.-C. Chen, C.-Y. Chang, A. Gavrielides, M. Camelo, J. M. Marquez-Barja, and N. Slamnik-Krieštorac, "Optimizing 5g network slicing: An end-to-end approach with isolation principles," in *2024 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*. IEEE, 2024, pp. 1–6.

[7] Z. Li, Q. Wang, Y. Wang, and T. Chen, "The architecture of ai and communication integration towards 6g: An o-ran evolution," in *Proceedings of the 30th Annual International Conference on Mobile Computing and Networking*, 2024, pp. 2329–2334.

[8] I. Chatzistefanidis, A. Leone, and N. Nikaein, "Maestro: Llm-driven collaborative automation of intent-based 6g networks," *IEEE Networking Letters*, vol. 6, no. 4, pp. 227–231, 2024.

[9] M. Abel, I. Ahmad, C. A. Casado, R. Berner, M. Bettinelli, K.-M. Björk, M. Capobianco, J. Gross, H.-T. Nguyen, P. Hui *et al.*, "Large language models in the 6g-enabled computing continuum: a white paper," 2024.

[10] C. University, "Edge ai service wrapper repository," https://github.com/Cranfield-GDP/edge-ai-service-wrapper, 2024, accessed: 2025-06-14.

[11] Y. T. et al., "Ai-ran simulator for edge ai orchestration," https://github.com/ntutangyun/ai-ran-sim, 2024, accessed: 2025-06-14.

[12] OpenAI, "Openai agents sdk," urlhttps://openai.github.io/openai-agents-python/, 2025, accessed: 2025-06-14.

[13] ——, "Gpt-4.1 model," urlhttps://platform.openai.com/docs/models/gpt-4.1, Apr. 2025, accessed: 2025-06-14.

[14] H. Face, "Models - hugging face," https://huggingface.co/models, 2025, accessed: 2025-03-11. [Online]. Available: https://huggingface.co/models

End-to-end edge AI service provisioning framework in 6G ORAN

Tang, Yun

2025-10-19

Tang Y, Srinivasan UC, Scott BJ, et al., (2025) End-to-end edge AI service provisioning framework in 6G ORAN. In: Proceeding of the 2025 IEEE 102nd Vehicular Technology Conference (VTC2025-Fall), 19-22 Oct 2025, Chengdu, China

https://doi.org/10.1109/vtc2025-fall65116.2025.11310512