PerceptRAN:
Towards Maturing O-RAN based Data Driven RAN Monitoring and Control

Open Networks Ecosystem Competition Project

Project Closure Report

Executive summary

The Project Closure Report document provides a comprehensive overview of the project's research and development efforts, technical advancements, and operational achievements. The primary aim of the project was to address critical limitations in the existing Open RAN (O-RAN) RIC ecosystem and enhance its readiness to support both current and emerging use cases. The project focused on several key areas, including the development of dynamic service models, the implementation of a real-time RIC (RT-RIC), and the creation of realistic training and testing environments for O-RAN applications.

One of the significant achievements was the development of dynamic service models that enable real-time coordination and control in applications such as interference mitigation and energy optimisation. The project also introduced a real-time RIC to support applications requiring sub-millisecond responsiveness and high adaptability. Additionally, the creation of large-scale emulation systems and testbeds provided developers with realistic environments to train and validate their algorithms.

The project also emphasised improving the overall experience for O-RAN developers by addressing challenges in the application development lifecycle and ensuring compatibility with multiple near real-time RICs. Furthermore, the project aimed to foster broader talent development and innovation within the UK by engaging with third-party developers and university students.

Overall, the project successfully addressed technical and operational bottlenecks in the O-RAN ecosystem, empowering developers and fostering innovation. The project's outputs, including open-source software and sample applications, have significantly contributed to the advancement of the O-RAN community and the broader telecommunications industry.

Aims and scope of the project

The primary aim of this project was to address critical limitations in the existing Open RAN (O-RAN) RIC ecosystem and to enhance its readiness to support both current and emerging use cases. O-RAN technology is fundamentally transforming how radio access networks (RAN) are managed and deployed by introducing programmable interfaces and enabling the integration of third-party applications (xApps and rApps). However, despite its immense potential, the ecosystem faces bottlenecks in standardisation, developer support, real-time capabilities, and the availability of realistic evaluation environments. Recognising these challenges, this project sought to pioneer critical advancements that would not only strengthen the technical foundation

of the O-RAN ecosystem but also empower developers and foster innovation in this space.

One of the key goals of this project was to address the slow standardisation of service models, which currently hinders the interoperability and usability of O-RAN xApps. By defining and implementing dynamic service models and building a programmable RAN platform compatible with O-RAN standards, we aimed to enable developers to create and deploy custom service models that could greatly expand the range of use cases supported by the RIC. These dynamic service models were designed to include codelets, lightweight and flexible components capable of dynamically altering service model behaviour. This programmability aimed to address the need for real-time coordination and control in applications like interference mitigation, energy optimisation, and latency-sensitive operations. The basic dynamic service model functionality was developed as a part of the pervious FRANC work. In this project we sought to harden the initial version and integrate it into a wider RAN ecosystem.

To complement the dynamic service models, the project sought to build on the initial design of a real-time RIC (RT-RIC) also done during the previous FRANC work. The RT-RIC was envisioned as a critical addition to the RIC ecosystem, enabling applications that require sub-millisecond responsiveness and high adaptability. The goal of the RT-RIC was not only to extend the capabilities of the existing near-real-time RIC (nRT-RIC) but also to introduce new architectural components to support innovative real-time use cases. These include integration of the RT-RIC with O-RAN's existing xApps architecture, as well as the introduction of dApps for the RT-RIC and APIs to ensure full interoperability. The design and implementation of these capabilities were central to our efforts in enabling advanced applications that push the boundaries of current RAN management capabilities.

Another key aim of the project was to improve the overall experience for O-RAN developers, addressing several challenges that have traditionally impeded the creation, deployment, and testing of RIC applications. We identified key gaps in the application development lifecycle and built tools to help developers overcome them. By improving flexibility and providing developers with better tools, we sought to streamline the process of integrating RIC applications into real-world deployments. Furthermore, we worked to ensure compatibility with multiple near real-time RICs to demonstrate the feasibility of deploying a diverse range of applications across various platforms.

To enhance the usability of the RIC platform, another goal of this project was to address the lack of a realistic environment for training and testing O-RAN applications – a challenge that has slowed progress in AI/ML-based RAN innovation. Developers have traditionally struggled to find meaningful data and realistic training environments to validate their algorithms. We aimed to change this by creating large-scale emulation systems for RAN that enable developers to simulate various deployment topologies and

traffic scenarios. These emulation systems were integrated into existing developer frameworks, acting as a reinforcement learning gym specifically tailored for RAN applications. By creating these environments, we provided the tools necessary for developers to train and refine their algorithms in near-real-world conditions without incurring the costs of physical infrastructure.

Additionally, to validate ML models and ensure that our solutions are grounded in real-world outcomes, the project sought to expand existing testbeds to include both indoor and outdoor environments. The indoor testbed, already developed during the FRANC project, encompassed a realistic private 5G deployment with 24 radio units (RUs) across six floors and over 50 user endpoints (UEs). This facility offered a controlled yet dynamic setup to evaluate a wide array of new RIC applications. To complement this, we set out to build a new outdoor testbed located at the University of Edinburgh. This outdoor environment allowed us to validate key insights gathered from the emulation systems and indoor testbed, offering a more diverse set of real-world conditions, including those typical of high-density deployment (HDD) scenarios. These testbeds provided the empirical foundation for refining dynamic service models, real-time control mechanisms, and developer tools, ensuring that the project outputs were both robust and highly applicable across diverse use cases.

Beyond technical objectives, we aimed for this work to provide a significant boost to the innovation ecosystem surrounding O-RAN. A central pillar of the project was the creation of sample applications designed to highlight the potential of the enhanced RIC architecture while demonstrating real-world applicability. These applications targeted a variety of high-value use cases that would resonate strongly with operator needs—use cases such as DAS, distributed MIMO, enhanced security protection, predictive maintenance, and others. By providing concrete examples of how the improved RAN platform could be leveraged, we aimed to reduce barriers to adoption and inspire further experimentation and investment by the wider O-RAN community.

The project also placed heavy emphasis on fostering broader talent development and innovation within the UK. By actively engaging with third-party developers and university students, we created opportunities for the next generation of technologists to directly contribute to advancing the O-RAN ecosystem. Our deployment of new applications in collaboration with these external stakeholders not only showcased the capabilities of the extended RIC platform but also facilitated skill-building in a rapidly evolving field. This approach was intended to widen the talent pool in the UK, enabling universities to better equip their students with practical experiences, and encouraging the creation of innovative startups, particularly in AI/ML-driven RAN solutions.

Finally, a broader mission of the project was to help bootstrap the O-RAN developer ecosystem by addressing its most pressing technical and operational bottlenecks. By introducing real-time RIC capabilities, standardising APIs, improving interoperability, and providing rich development environments, we aimed to empower a diverse range of developers—from individual students to enterprise solution providers—to participate in and contribute to the O-RAN ecosystem. A particular focus was placed on enabling new and transformative use cases that had previously been inaccessible due to technical limitations or inadequate developer support. This work not only sought to accelerate adoption of O-RAN technology but also to solidify its position as a viable and scalable framework for modern RAN management.

Results

Distributed AI platform architecture: As a part of the project, we developed several components. Each of them addressed different components on the initial scope. We start this section by describing a distributed AI-native RAN platform whose goal is to address the challenges O-RAN developer face when writing AI-based applications. Recent advances in large language models and AI agents upended the future of software development. The goal of the distributed AI platform is to offer a new, simplified process of writing O-RAN applications based on AI.

One challenge when writing AI applications is the lack of access to the right features. Current O-RAN architectures typically rely on standardised APIs that expose coarse-grained information such as basic metrics or aggregated counters, which are often too simplistic to capture the nuanced dynamics needed for advanced machine learning algorithms. Furthermore, because data is often collected in isolated silos at different layers of the network (e.g., PHY, MAC, or application layers), the inability to aggregate and align these data streams in a meaningful way limits the capacity of AI models to generate insightful predictions or optimise RAN operations effectively.

The second challenge with the current O-RAN architecture imposes a hard split between xApps and rApps, forcing developers to work within rigid boundaries. These boundaries may vary among deployments. For example, one deployment may have C-RAN edge with a lot of GPU compute next to vDUs and another deployment may have GPUs only in a centralised edge, next to a vCU. Different customers may desire to deploy different applications, utilising the available GPU and CPU compute capacity in different ways. A developer needs a simple way to handle these deployment differences without rewriting the entire applications.

We started by proposing a holistic architecture to address these challenges, composed of dynamic service models, a real-time RIC, an interface to the existing O-RAN

components (nRT-RIC), and a centralised orchestrator that help developers deploy their workloads across different deployments with minimal changes.

Dynamic service models: The dynamic service model layer introduced significant methodological innovations. Conventional RAN systems often expose data through coarse-grained and static interfaces, limiting their usefulness for AI applications that thrive on granular, context-specific input. To overcome this limitation, we developed and deployed programmable data probes that are inserted dynamically at various points in RAN functions and network interfaces. These probes allow the selective collection of data tailored to the specific requirements of individual AI workloads, thereby reducing both the volume of transmitted data and the latency associated with preprocessing and aggregation. By decoupling data collection from static APIs and enabling it through programmable probes, we achieved a level of adaptability that enhances the system's potential to operate optimally in diverse scenarios.

Real-time RIC: The AI processor runtime represents another critical infrastructural innovation. This runtime abstracts the heterogeneity of different hardware and RAN configurations, presenting a uniform execution environment for AI applications. It allows the same code to run withing different type of RIC products, thus adapting to different deployment requirements.

A prototype implementation of the AI runtime is particularly suited for a real-time RIC. It introduced significant advancements to far-edge processing. By leveraging lightweight inference frameworks that can operate with minimal overhead on CPUs, the runtime addresses the low-latency and real-time constraints inherent to RAN-specific applications. We demonstrated that optimised components of the runtime, deployed on edge servers with low compute power, could efficiently process complex AI workloads such as interference management and predictive maintenance without compromising on response times.

This allowed us to provide a fully functional implementation of a real-time RIC concept. There has been a lot of discussion recently in the O-RAN around real-time use cases and applications that are not supported well but the current O-RAN architecture. These have been recently summarised in the O-RAN NGRG RR-2024-10 report.

The use cases can be loosely classified in two groups. One group are applications that require real-time interaction with the RAN stack. One such example is real-time packet scheduling. Many industrial and vertical use cases have hard latency constraints for certain classes of traffic and may require customised packet scheduling. Scheduling operations must happen at the time scale of RAN slots (< 1ms), which is an order of magnitude faster than what the near-RT RIC design supports.

The other group are applications that perform inference based on user plane data. One such example is interference classification. This application is required to process

received IQ samples at the front-haul rate of over 1Gbps. Shipping all the samples to a near-RT RIC, typically deployed at a central location, can be prohibitively expensive. Furthermore, any data serialisation and deserialisation introduce communication and compute overhead which is especially taxing at such high data rates.

We designed and developed a real-time controller to address these requirements. The real-time controller allows application developers to develop and deploy a new class of real-time applications, often referred to as dApps. The controller and the applications run in a separate process on the same server as RAN. They communicate with a RAN through the dynamic service models.

The real-time controller is tightly integrated with dynamic service model using shared-memory, zero-copy communication. This allows it to achieve sub 10us communication latencies with RAN. The zero-copy design implies that dApps can safely access RAN data directly from RAN memory, with no copy or serialisation overhead, addressing the second of the two requirements.

Integration with E2 protocol: We further implemented an integration with the existing O-RAN nRT-RIC component through the E2 interface. This option is recommended in a recent O-RAN NGRG RR-2024-10 report. The E2 interface is a generic interface whose data semantic is specified through service models. We use a new, dynamic O-RAN E2 service model. This model supports a load and unload control messages that can load a number of codelets, dApps, and data schemas. It also supports uplink and downlink data messages that send data described by the uploaded schemas to and from the RAN.

This dynamic service models brings the flexibility to the application developers to manage all components of the application from an existing xApp through standard nRT-RIC management interfaces. Each xApp is responsible to manage the life cycle of the dApps, codelets and service models associated with it. This allows a seamless integration of the new components in the existing O-RAN architecture.

Building on these foundational components, we developed an initial prototype of an integrated orchestrator, which serves as the intelligent control plane responsible for optimising the placement and lifecycle management of AI applications across distributed infrastructure. This component dynamically mapped applications to runtime instances based on developer requirements (e.g., latency thresholds, privacy considerations, compute requirements) and infrastructure constraints.

We further developed several applications that illustrate the benefits of different components. These applications also address some of the critical challenges of the existing O-RAN deployments, such as efficiency, security and operational challenges.

Front-haul attack detection: The adoption of Ethernet-based protocols like eCPRI in the fronthaul opens up vulnerabilities to security risks such as packet manipulation and man-in-the-middle (MITM) attacks due to the lack of mandatory integrity protection. These software-based MITM attacks can be devastating, causing widespread performance degradation and denial of service even without a physical radio presence. While proactive measures like MACsec offer protection, they are not yet universally implemented. Therefore, reactive measures like anomaly detection in real-time radio access network (RAN) telemetry provide a practical interim solution to identify and mitigate these threats. Real-time RAN anomaly detection tools, such as Janus and RT-RIC, enable continuous monitoring of metrics, allowing detection of irregular patterns indicative of MITM attacks, triggering alerts, and facilitating remediation efforts.

Anomaly detection methods leverage predictable traffic patterns in control-plane (C-plane) and user-plane (U-plane) packets, such as consistent inter-packet delay ranges that align with fronthaul specifications. Any deviations in these patterns can be flagged as anomalies, although they may sometimes arise from benign issues like hardware faults or network contention. To improve accuracy, these traffic deviations can be correlated with higher-layer Key Performance Indicators (KPIs). For example, attacks like FRONTSTORM (A1 and A2) or payload corruption (A3) can be identified by linking anomalies in fronthaul traffic patterns with system-level spikes in signalling messages, CQI, or BLER changes. This multi-layered detection approach significantly enhances security until proactive measures like mandatory MACsec become standardised.

Improve RAN efficiency through coordination: Smaller vendors and third-party developers still face significant barriers when attempting to meet the RAN efficiency achieved by leading vendors. These include challenges in delivering capabilities like interference mitigation, massive MIMO, and resource sharing, all of which are typically reliant on expensive, proprietary solutions or dedicated hardware from dominant players. Additionally, restricted access to full RAN data and control interfaces—often held under proprietary implementations by incumbents—further hampers third-party innovation and competition, limiting the realisation of Open RAN's full potential.

To address these challenges, we developed a software-based middlebox architecture for the RAN fronthaul—the communication interface between the RU and DU. The middle-box intercepts control-plane (C-plane) and user-plane (U-plane) traffic to enable advanced features and applications without requiring modifications to the existing RAN infrastructure. By leveraging the standardised, vendor-agnostic nature of Open RAN fronthaul protocols and utilising widely adopted networking technologies like DPDK and XDP, the middle-box transparently introduces capabilities such as real-time packet manipulation, payload inspection, and data caching.

We implemented several applications to demonstrate the potential of this approach, such as a digital Distributed Antenna System (DAS), a distributed MIMO (dMIMO)

through logical mapping of physical RUs, RU sharing between multiple operators and real-time monitoring of spectrum utilisation. All these are built without any changes to the existing RAN systems. Developers can build more extend functionalities through apps, leveraging the dynamic service models and real-time RIC.

Cloud based RAN emulation system. There are multiple reasons that motivate the need for controlled testing environments for RAN that are realistic and scalable. This includes: (1) data generation for training and testing of AI/ML based Apps; (2) realistic at scale evaluation (e.g., at city scale); (3) sandbox environment to test new features and innovations prior to deployment in production networks. For this purpose, emulators provide the right middle ground between simulators and real testbeds. Existing RAN emulation solutions, however, are too expensive, do not scale or compromise fidelity. Hardware based RAN emulators (e.g., Viavi TM500) are expensive to scale. Software based emulators are relatively promising but current approaches either trade off fidelity (e.g., ns-3) or scale (e.g., EMANE).

Motivated by the above, we propose a new approach to leverage elastic and commoditised cloud compute for cost-effective, high-fidelity and scalable RAN emulation. While this is a compelling proposition, running RAN emulation scenarios in a public cloud setting and at large scale presents new challenges. First, the RAN functions have high requirements in terms of compute and network resources, that can reach several CPU cores and gigabits of traffic per cell. Second, the RAN software is required to respect stringent sub-millisecond latency requirements (e.g., for the scheduling of radio resources). But the public clouds have not been designed with such (near) real-time requirements in mind.

To overcome these challenges, we have developed a cloud-based RAN emulation system design called Chronos, whose schematic is provided in the figure below. Chronos brings together multiple key ideas to enable RAN emulation at any scale: i) it replaces the compute and network intensive PHY layer and channel of the RAN and associated user devices (UEs) with an emulated link based on the standard FAPI interface, ii) it introduces a custom hypervisor that virtualises the emulation time, effectively shielding the emulated network functions from external CPU and network latencies, iii) it operates at the granularity of RAN slots and uses the slots as synchronisation barriers among emulation components to allow flexible scaling, and iv) it introduces a logically centralised software switch to forward the control and data plane FAPI traffic between the RAN and the emulated mobile devices in a scalable manner.

Campus scale private 5G Open RAN testbed in Edinburgh. Campus-scale private 5G Open RAN testbeds provide a critical environment for advancing next-generation mobile

networks. They enable experimental validation of machine learning-based Open RAN applications at scale, offering real-world data for digital twin validation and optimisation. These testbeds support the automation and optimisation of RAN operations, covering aspects like energy usage, spectrum management, TDD, CA, and handovers. By integrating AI at the edge, these testbeds enhance performance beyond traditional RAN, enabling intelligent, low-latency applications. Additionally, they facilitate the investigation of multi-vendor interoperability and address operational issues, fostering the development of flexible, efficient virtualised RAN systems.

Moreover, campus testbeds allow for studying disaggregated and distributed core deployments, where the data plane operates at the edge and the control plane is in the cloud. They also support seamless interoperation between private and public mobile networks, optimising service quality and device-side experiences over private 5G networks. These testbeds are invaluable for pushing the boundaries of 5G technology, ensuring robust and efficient solutions for a wide range of use cases, from enterprise applications to advanced connectivity.

Building on the need for real-world 5G testing, we have deployed a private 5G Open RAN testbed at the Edinburgh campus, offering a comprehensive and scalable platform for advanced mobile network research. This testbed features a campus-scale private 5G deployment in a high-density urban environment, ensuring a realistic setting for testing and optimisation. It is O-RAN compliant and designed to be flexible, supporting a disaggregated architecture with multi-vendor components, including radio units (RUs) from two different vendors. The testbed incorporates both open-source and commercial virtual RAN (vRAN) software, along with multiple different RAN Intelligent Controllers (RICs), enabling a wide range of configurations and use cases.

Additionally, the testbed includes a cloud-based core and is equipped with basic AI compute at the edge, allowing for the exploration of edge computing and AI-driven network optimisations. It supports mobility and works seamlessly with a variety of commodity devices, offering connectivity via physical or eSIMs. This flexibility makes the testbed an invaluable resource for studying diverse network scenarios, ensuring that both the operational and user experience aspects of private 5G are thoroughly tested and refined.

We have also carried out extensive coverage analysis for this testbed both through simulation and real-world measurements. For the simulation, we considered two different tools – CloudRF (empirical-channel-model-based) and SionnaRT (ray-tracing-based).

Methods for efficient Open RAN telemetry. The potential efficiency and automation gains from data-driven operation in Open RAN are determined by the scope and

granularity of KPIs – “the data”. Fine-grained KPI data is generally better than coarse-grained data, although desired granularity is KPI dependent. Fine granularity KPI data collection, however, causes high overhead and can disrupt the RAN operation. Also note that the capability to measure any KPI (e.g., Janus) is necessary but insufficient. This leads to the following challenge: can we collect KPI data at desired granularity without incurring the overhead?

To address the above challenge, we have first developed a generative AI based solution called NetGSR (schematic in the figure below). NetGSR is made up of two components: (1) DistilGAN for KPI data stream reconstruction to its original resolution at high fidelity from a low-resolution version; and (2) Xaminer for adapting the sampling rate at the measurement data source (e.g., DU function). We have evaluated the effectiveness of NetGSR approach considering a wide range of Open RAN KPIs collected from the Microsoft Cambridge testbed. Our results show that NetGSR yields 20x efficiency gain with negligible loss in fidelity.

NetGSR, however, is ineffective for physical-layer telemetry, especially for IQ samples, which are complex-valued, multivariate, and structured across time, frequency, space, and code domains. Their characteristics are also sensitive to environmental factors like multipath and interference, making reconstruction from coarse data unreliable. We further find that existing IQ compression methods also are limited in the efficiency gain they can offer before fidelity is compromised.

To address the limitations of existing IQ compression techniques, *the proposed method TAPS (Three-Dimensional Amplitude-Phase-Spatial Compression) method* introduces a structured and efficient solution tailored to the characteristics of physical-layer telemetry in O-RAN. Unlike prior approaches that treat IQ data as unstructured or assume sparsity in fixed domains, TAPS is designed to exploit structured correlations across multiple dimensions: amplitude and phase, spatial antenna ports, time-domain OFDM symbols, and frequency-domain resource elements. This approach allows TAPS to achieve high compression ratios—up to 32×—with minimal information loss.

Unsupervised anomaly detection method for Open RAN. The complexity of Open RAN systems presents significant challenges in troubleshooting RAN related performance issues and failures. These challenges mainly are centered around the ability to perform accurate anomaly detection with Open RAN KPI data. We find that conventional methods often result in unacceptable false detections when applied as is in the Open RAN context. So, in the earlier FRANC project, we proposed Spotlight, a generative AI based solution to address this issue. Despite the fact that SpotLight introduces a tailored anomaly detection and identification pipeline for Open RAN setting and represents the new start-of-the-art for anomaly detection in this domain,

our experience highlighted a major impediment to its effectiveness in real-world deployments that is linked to how it is trained.

Like prior best performing anomaly detection methods, SpotLight also falls under the class of semi-supervised methods which require labelled normal data for training. As what is “normal” for KPI data in an Open RAN setting is incredibly diverse and complex, influenced by a variety of factors including user traffic patterns, device mobility, wireless channel and interference conditions, platform related aspects, etc. Accordingly, it is practically impossible to gather the corpus of normal data for training that is sufficiently representative of all possible scenarios that can occur in practice. Ultimately, the generalisability and real-world effectiveness of semi-supervised methods like SpotLight hinges on how well all possible types of normal data are covered in the training data.

In this project, we set out to remove the requirement for having any labeled data for training. In other words, we seek an anomaly detection method for Open RAN that performs as well or better than SpotLight *but without requiring normal data to train it*. In machine learning terms, this translates to designing an “unsupervised” anomaly detection method. This is however a significant challenge as the literature shows; all state-of-the-art anomaly detection methods within Open RAN and beyond follow a semi-supervised paradigm and existing unsupervised methods can only offer inferior anomaly detection performance.

To overcome this challenge, we have come up with a new insight that when training a model for some learning tasks on a mix of normal and anomalous data, the model converges faster on normal data while struggling to fit to anomalous data. Armed with the above insight, we have developed NARCISSUS, a new unsupervised anomaly detection method for Open RAN and beyond. NARCISSUS leverages training dynamics for accurate and robust anomaly detection with unlabelled data through a combination of a tailored early stopping algorithm and an ensemble method.

Our comprehensive evaluation results demonstrate the promise of NARCISSUS to successfully turning an anomaly detection algorithm originally designed for semi-supervised anomaly detection (e.g., SpotLight) to an unsupervised setup while maintaining or often improving accuracy.

Open RAN sensing applications with a focus on device positioning. We observe that the NextG radio network can be double up as a sensing infrastructure. The emergence of Open RAN and its likely adoption in NextG networks is a significant and timely contributing factor to this case. With Open RAN, all the functionality realised previously inside the RAN components can now be realised as an ``App'' over the RAN Intelligent Controllers (RICs) using the data from the RAN. Furthermore, each of those Apps can be

innovated leveraging the power of AI. This paradigm shift in the RAN architecture not only allows rethinking RAN operation aimed at resource allocation and network management for improved efficiencies and greater automation but also opens the door for imagining new types of data-driven RAN Apps that are not limited to improving RAN operations. As a result, we can envision a suite of sensing related Apps focusing on sensing of various kinds (including positioning) that are enabled by data from the underlying RAN infrastructure.

We focus on the device positioning use case – a key application in the realm of joint communication and sensing (JCAS) for 5G and beyond. Positioning plays a critical role in enhancing the performance of both applications and networks by enabling more precise mobility awareness. This awareness allows for optimised applications like improved streaming and more efficient mobility management at the Radio Resource Control (RRC) layer. Additionally, positioning facilitates better scheduling, channel quality prediction at the Medium Access Control (MAC) layer, and enables advanced MIMO beamforming at the physical layer, all of which are essential for enhancing network efficiency.

Positioning is also vital for emerging applications like industrial automation, Vehicle-to-Everything (V2X) communications, and smart city infrastructure. As a network-centric function, device positioning is inherently data-driven, making it a prime candidate for machine learning (ML) techniques. These methods can further enhance positioning accuracy and efficiency, contributing to innovations in network-side drive testing and the broader deployment of next-generation technologies.

We have investigated the positioning use case in two different settings. First, we considered line of sight positioning of devices indoors based on 5G reference signal measurements from the Cambridge indoor carrier-grade 5G Open RAN testbed. This case study demonstrates how communication reference signals can be leveraged to enable sensing functionalities as well as highlights the associated challenges. Recently we switched focus to the more complex outdoor and non-line of sight case considering the Edinburgh outdoor campus testbed. In both cases, we highlighted the potential for leveraging machine learning techniques to overcome the various practical challenges.

Adversarial ML attacks on Open RAN Apps. Unlike traditional RAN, the O-RAN architecture enables access to RAN data (i.e., network telemetry) via RAN intelligent controllers (RICs) to third-party machine learning (ML) powered applications – such as, rApps and xApps – to optimise RAN operations. Consequently, there is currently tremendous amount of focus on leveraging RAN data to unlock greater efficiency gains. However, there is an increasing recognition that RAN data access to Apps could become a source of vulnerability and exploited by malicious actors.

Motivated by this, we first conduct a comprehensive qualitative examination on the relative feasibility of adversarial attacks targeting ML based Apps in O-RAN systems. Our analysis suggests that black-box evasion attacks based on inference data access are the most viable and identify ways that internal/external adversaries can launch such attacks. Informed by this, we design a novel black-box evasion attack strategy that comprises four key techniques, namely, model cloning algorithm, input-specific perturbations, universal adversarial perturbations (UAPs), and finally, targeted UAPs– which together are aimed at attacking Apps within O-RAN system and degrading network performance.

With a particular focus on rApps over non-RT RIC, we experimentally validate the effectiveness of the designed evasion attack strategy and quantify the scale of performance degradation leveraging a commercial emulation environment. Further, we show that proposed attack strategy is effective against prominent defence techniques for adversarial ML such as defensive distillation and adversarial training.

Project costs

For Microsoft, most of the cost was in software development. They already funded our large-scale indoor 5G testbed with DSIT FRANC project funding. During the ONE project, they focused on software development on top of the existing hardware and software infrastructure.

Additional development cost went into the testbed management. They needed to deploy software to manage RAN and client UEs remotely, to execute various experiments. This allowed them more realistic use cases, and more flexibility compared to industry standard testing solutions.

For University of Edinburgh, majority of the cost went into the outdoor campus testbed deployment and associated personnel costs. Remainder of the costs were for research personnel who worked on other components of the project that Edinburgh led including cloud-based RAN digital twin system and various machine learning based Open RAN apps including positioning and anomaly detection.

Impact and benefits

There are several tangible benefits from this project. Firstly, most of the software developed as a part of the project is now open sourced. This includes the key components such as dynamic service models (jbpf) and real-time RIC (jrt-controller). This allows the entire O-RAN community to immediately access them, study and potentially integrate with their solutions.

In addition to integrating with one of the partner's Capgemini 5G RAN, we also integrated the components to open source srsRAN project. This project is popular with academic communities. We are also providing various sample applications as open source, to facilitate community adoption. Furthermore, we are working with several universities directly to help them onboard on the platform. For example, we work with University of California San Diego to help them integrate their EdgeRIC research platform on top of our software. We are starting similar activities with several other universities.

In parallel, we worked with various vendors to help them evaluate and integrate our deliverables into commercial platforms. So far, we made the biggest progress with Mavenir, one of the leading O-RAN providers. Mavenir has tested our dynamic service model over several years. The design passed their strict performance testing, and they have now integrated it into their product. The latest version of the product is now being deployed in the networks of their commercial, tier-1 customers. Mavenir is now interested in integrating other parts of our overall architecture, and in particular the real-time RIC design.

We further successfully tested our E2 integration with two different RIC. One is an open source FlexRIC. The other one is Juniper nRT-RIC product. We also worked with various startups to help them build their applications on top of our stack. We made most progress with a localisation company Zainar.

We also had extensive presence at various industry events. We had multiple demos at Mobile World Congress, PRs and blogs, and technical white papers. We demonstrated our software platform to several major tier-1 operators by trialling in their labs and responding to their questions. All these activities help spread the awareness of our approach.

Learning from the project

The collaborative efforts of academic and commercial partners highlighted differences in internal finance systems and hiring constraints, revealing that planning projects to start immediately after Grant Funding Agreement (GFA) approval may not be practical for all stakeholders. Additionally, GFA approvals – even for project continuations – were more time-consuming than expected, emphasising the need for early engagement of all parties to streamline the process. For outdoor testbed setups, significant delays arose from factors such as GFA processes, evaluation licenses, RU hardware procurement, and mast installation timelines. Addressing these challenges requires centralising key processes (e.g., evaluation licenses and RU procurement) and leveraging standardised practices, such as a certified list of mast installers. These enhancements would facilitate smoother multi-vendor Open RAN deployments, avoiding setbacks like missing AI compute integration at the edge.

From a technical standpoint, initial findings revealed that operators currently favour Non-Real-Time over Near-Real-Time RIC, necessitating efforts to adapt work to different architectures and use cases. Simplification and modularity proved critical in achieving the security levels required for dynamic service models, as eliminating unnecessary libraries reduced vulnerabilities and allowed vendors to balance features with security priorities. Furthermore, AI-based RIC applications exposed gaps in existing architectural abstractions and APIs, indicating a need for new designs better suited to edge AI requirements. To address these issues, engaging the broader AI and edge computing ecosystem proved beneficial by unifying efforts across industries and raising early awareness of standardisation needs. Telecom vendors also exhibited interest in broader AI use cases beyond RAN, suggesting the need for a holistic approach to edge AI integration and monetisation strategies across carriers and infrastructure investments.

Another significant insight focused on the programmability of fronthaul interfaces, demonstrating that advanced RAN technologies like Distributed Antenna Systems (DAS)

and Distributed MIMO (dMIMO) can be implemented cost-effectively in software rather than hardware. This finding expands the usability of Open RAN beyond traffic management and optimisation to innovative applications via programmable interfaces. These lessons collectively underscore the importance of modular system designs, early industry engagement, cross-disciplinary collaboration, and efficient project management to address technical, operational, and ecosystem-wide challenges in advancing Open RAN initiatives and edge AI applications.

Conclusion

The Project Closure Report highlights the significant advancements made in the Open RAN (O-RAN) RIC ecosystem. The project successfully addressed technical and operational bottlenecks, empowering developers and fostering innovation. Key achievements include the development of dynamic service models, the implementation of a real-time RIC, and the creation of realistic training and testing environments for O-RAN applications. The project's outputs, including open-source software and sample applications, have potential to significantly contribute to the advancement of the O-RAN community and the broader telecommunications industry.