Detection of UE Metric Data Poisoning in O-RAN: Automated Risk Analysis and Resilience

Mark Megarry, Vishal Sharma, *Senior Member, IEEE,* Berk Canberk, *Senior Member, IEEE,* and Trung Q. Duong, *Fellow, IEEE*

***Abstract*—The implementation of open radio access network (O-RAN) architecture brings with it new security challenges, owing to the use of machine learning (ML) methods on the near-real-time radio access network (RAN) intelligent controller (Near-RT RIC) and non-real-time RAN intelligent controller (Non-RT RIC). Some of the threats arise from large-scale data poisoning. Considering this, a novel system of security solutions, deployed as xApps and rApps in the O-RAN network, which seeks to address the threat posed by an adversary which targets the network utilising a large botnet of user equipment (UEs) for the purpose of data poisoning is proposed. This security system comprises a UE risk analysis xApp running on the Near-RT RIC, an xApp misbehaviour detection function running on the Near-RT RIC platform, and a resilience management rApp running on the Non-RT RIC. An experiment is presented which illustrates the operation of the risk analysis system's UE classifier and the effects of varying an associated hyperparameter, which re-emphasises careful tuning to optimise the classifier's performance and, consequently, risk analysis accuracy.**

***Index Terms*—Data poisoning, O-RAN, Security, Misbehaviour detection, Resilience.**

I. Introduction

Open radio access network (O-RAN) is a RAN architecture which supports network function virtualisation, open interfaces between network components, and closed-loop, artificial intelligence (AI)-enabled control for network security and optimisation, as described in technical specification of O-RAN Working Group 1 (WG1.OAD R003 v12.00) [1]. If O-RAN architecture is to be an enabling technology of sixth-generation (6G) terrestrial and non-terrestrial [2] wireless networks, it must ensure the availability of RANs and guarantee the confidentiality of information passing through them, while maintaining acceptable latency and throughput performance [3]. Based on these requirements, this work investigates methods for detecting ML model data poisoning attacks executed by manipulating UE key performance indicator (KPI) reports. The methods considered include a UE risk analysis system, an xApp misbehaviour detection system, and a resilience management system. We use the following definitions for this work:

- Resilience: The ability of a system to mitigate and recover from attacks in a timely manner.
- Survivability: The ability of a system to remain operational in the event of an attack.
- Risk: The probability of an attack occurring, weighted by the severity of the outcome of that attack.
- Threat: An event which negatively impacts the system.

A. O-RAN Threat Modelling

O-RAN Alliance Working Group 11 (WG11) have identified potential threats against O-RAN network components, and have described models of threat agents in their threat modelling and risk assessment technical report (WG11.Threat-Modeling.O-R004-v04.00) [1]. These agents are cyber-criminals, insiders, hacktivists, cyber-terrorists, script kiddies, and nation-state actors [1]. The categories of threats against O-RAN networks identified by WG11 include threats against O-RAN system, threats against O-Cloud, threats to open source code, physical threats, threats against 5G radio networks, threats against AI/ML systems, protocol stack threats, SMO threats, and threats against shared O-RU [1]. This work seeks to address data poisoning attacks, which WG11 consider as a subset of threats against AI/ML systems [1].

B. Adversarial Machine Learning and O-RAN

Adversaries targeting O-RAN networks may attempt to utilise adversarial machine learning (AML) techniques to attack ML models running on the Near-RT RIC and Non-RT RIC [4]. Habler et al. have described the AML threat categories of evasion, model corruption (including training data poisoning attacks), membership inference, data property inference, data reconstruction, model extraction, and resource exhaustion in O-RAN [4].

1) ML Pipeline and Data Poisoning: O-RAN Working Group 2 have specified a framework for the deployment and training of ML models in their technical report (WG2.AIML v01.03) [1]. This framework supports the continuous acquisition of network data via the O1, A1, and E2 interfaces, allowing data to be transferred from the O-RAN central unit (O-CU), O-RAN distributed unit (O-DU), Near-RT RIC, and, Non-RT RIC [1]. This continuous data acquisition enables models running on the network controllers to continuously learn as data is produced by the network, however, it also introduces a vector through which data poisoning attacks may be carried out [5]. An attacker with access to nodes communicating via these interfaces may seek to influence the data produced from them, manipulating the training data of ML models and potentially changing their behaviour to the attacker's benefit [5]. Fig. 1 illustrates an example inspired by the QoS-Based Resource Optimisation use case provided by O-RAN Work Group 1 [6], which is also considered in work [7]. In this example, a quality of service (QoS) optimisation xApp running on the Near-RT RIC utilises an ML model with RAN KPI reports as training data, and must allocate

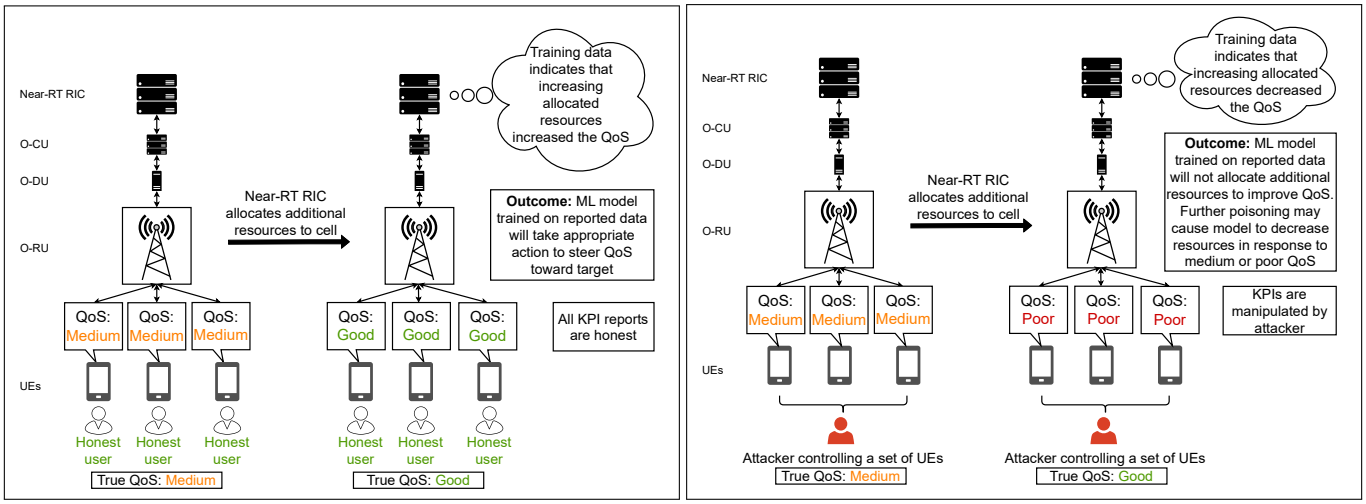

Fig. 1: Example QoS optimisation scenario (a) with only honest users. (b) with attacker present carrying out data poisoning.

resources to optimise the QoS experienced by UEs attached to an O-RAN radio unit (O-RU). In our example, the true QoS for the UEs is initially "medium". In response to this, the xApp allocates additional resources to the cell in an attempt to increase the QoS. In scenario (a) with honest users, the QoS reports will change from "medium" to "good", and the training data will then indicate that increasing the allocated resources improved the QoS. In scenario (b), the attacker controls a large set of UEs and may manipulate the QoS at those UEs, possibly by causing intentional interference via other equipment or by manipulating the behaviour of the UEs. When the xApp allocates additional resources to the cell to improve the attacker's QoS from "medium", the attacker will cause the QoS to appear as "poor". The resulting training data will incorrectly indicate that increasing the allocated resources decreased the QoS. This introduction of malicious data to the ML training dataset is known as data poisoning, which is distinct from similarly named attacks e.g., the bearer migration poisoning attack [8], which manipulates traffic paths in O-RAN networks.

2) Impact of Model Corruption Attacks in O-RAN: ML model corruption may affect the integrity and availability of the ML model [4], however, the impact of such an attack in O-RAN networks would depend on the model which is corrupted. Considering a model used to control traffic steering [6], an attack affecting the model's decisions could potentially result in excessive amounts of traffic being steered to a targeted node, causing a denial of service. Considering a model which is used for vehicle-to-everything (V2X) handover optimisation, perhaps model corruption could result in ping-pong handovers [6], affecting the quality of service. Corruption of a model responsible for optimising the QoS and energy consumption of O-RUs could possibly result in poor QoS from affected O-RUs, which may cause denial of service.

C. Technologies Supporting Security in O-RAN

This section introduces zero trust architecture, anomaly detection (AD), and threat intelligence. These technologies may be used to counter the emerging threat posed by attackers utilising and targeting AI/ML technologies.

1) Zero Trust Architecture: The O-RAN Alliance are utilising zero trust architecture and principles in the development of O-RAN [1]. According to NIST, zero trust principles assume that an adversary is present on the network, therefore, no party is implicitly trusted, access to resources is limited to parties requiring access, and the identities of parties are continuously authenticated [9]. Zero trust architectures are built on these principles, seeking to limit the lateral movement of adversaries through the network [9].

2) Anomaly Detection and Threat Intelligence: AD is a task in which unusual, outlying or unexpected data points are identified [10]. In the context of O-RAN, AD may be used to identify unusual behaviour or patterns in UE KPI reports. The O-RAN Software Community (SC) maintains an xApp which uses the Isolation Forest algorithm to detect anomalies in UE data [11]. This xApp receives UE and O-DU metrics, including reference signal received power (RSRP), reference signal received quality (RSRQ), and throughput [11]. Regardless of whether the response to a detected anomaly is carried out by a human or an automated system, it is usually desired to minimise the number of alerts produced by an AD system to benign data, or to categorise the alerts by severity. For human operators, analysing a large number of alerts may increase the time taken to respond to a security incident, or may result in the operator overlooking important information in a phenomenon known as alert fatigue [12]. In automated response systems, it follows that a high rate of alerts to benign data may result in an increased time taken to respond to security incidents. AD techniques may be enhanced by threat intelligence, which can be considered as the use of gathered information to identify

and anticipate threats [13]. While many AD techniques do not require information regarding the behaviour of existing threats to effectively detect anomalous behaviour, it can be advantageous to utilise this information. For example, the Cluster Centers algorithm proposed by Castellani et al. [10] examines the distance between a data point and the centre of a cluster of "normal" data, as well as the distance between the data point and labelled anomalous data points in order to derive an anomaly score. This concept may be applied to the domain of security, with labelled data corresponding to known attacks being considered in the detection of anomalies.

II. Solution and Contributions

Given the threat posed by an attacker with access to a large botnet of UEs, who seeks to carry out data poisoning attacks on ML models running on the Near-RT RIC and Non-RT RIC, how can these attacks be detected and mitigated? This article proposes a UE behaviour monitoring system, which aims to identify unusual RAN metrics in subscribers. The UE monitoring system derives a risk score for each UE based on the behaviour of the UE, and the expected behaviour for UEs in the category of the device under test. In Section VI, it is discussed how this solution may be extended to overcome an enhanced threat model, which requires the UE risk analysis system to be monitored for unusual behaviour and redeployed (possibly in another location in the network) if it is compromised. The proposed solution to address the enhanced threat model features three subsystems: the UE risk analysis system, the xApp misbehaviour detection system, and the resilience management system.

The contributions of this paper include,

- A novel UE risk analysis framework which identifies the use case of the UE under test and considers the true behaviour of the UE relative to the expected behaviour for its use case.
- Demonstration of the UE risk analysis system's use case classifier operating on 5G New Radio data provided by Ofcom [14].
- Details on open research problems which must be addressed prior to the implementation of the risk analysis framework, including addressing the threat posed by malicious xApps, and identifying which KPIs should be used for this application.

III. Threat Model

This work considers an adversary, with access to a large botnet of UEs, who seeks to carry out data poisoning attacks on an O-RAN network. The UEs in the attacker's botnet may have been registered to the network by the attacker with the intent of carrying out an attack, or they may have initially been in the possession of honest users and later infected with malware. This attacker attempts to carry out an indirect data poisoning attack, in which the botnet of UEs is utilised to generate false data in order to manipulate ML model training through the manipulation of KPIs. An attack is considered as being active or inactive. While the attack is inactive, all UEs in the attacker's botnet behave "normally", producing KPIs according to their application. While the attack is active, the behaviour of some or all of the UEs in the attacker's botnet will change, producing KPIs different from when the attack is inactive. If the attacker is acting stealthily, then the difference between the KPIs produced while the attack is inactive and those produced while the attack is active will be minimised while still resulting in the desired goal.

IV. UE Risk Analysis System

The proposed UE risk analysis system is capable of analysing RAN KPIs, as reported by a KPI monitoring xApp, in order to classify UEs into pre-defined behaviour groups (e.g., industrial internet of things, sensor station, media streaming, web browsing, V2X, or airborne/drone). The risk posed to the network by each UE is found based on the classification of the UE, and the UE's behaviour relative to its group. The data input to this system includes UE RSRP, RSRQ, and signal to interference and noise ratio (SINR). This solution also implements elements of threat intelligence, as discussed in Section I-C2. Similar to the Cluster Centers method proposed by Castellani et al. [10], the proposed solution examines the distance from a data point to the mean values for its group, and the values corresponding to known attacks. This section details existing work carried out to detect unusual behaviour in UEs, potential improvements to this existing work, and the proposed method for analysing the risk posed by each UE connected to the network.

A. Existing Work and Potential Improvements

As discussed in Section I-C2, the AD xApp maintained by the O-RAN SC utilises the Isolation Forest algorithm to detect anomalies in UE data [11]. This method, however, does not necessarily utilise knowledge of expected behaviour for the specific type of the UE. Our risk analysis system seeks to provide an alternative approach and reduce unnecessary alerts, by comparing UE data to the expected behaviour for the UE's use case, rather than utilising training data gathered from all UEs used in various applications. In this way, our system considers that data expected for one UE group may be unusual in another UE group. E.g., throughput patterns generated from a mobile phone being used for web browsing may be unusual to see coming from a sensor station, and vice versa. We propose that the UE risk analysis system be implemented as an xApp. As such, it may receive RAN KPI data aggregated by a KPI or traffic monitoring xApp and enrichment data from the Non-RT RIC via the A1 interface to augment its risk analysis capabilities.

B. Method

The UE risk analysis method consists of three steps: Training, classification, and risk scoring. The interactions between these steps are illustrated in Fig. 2.

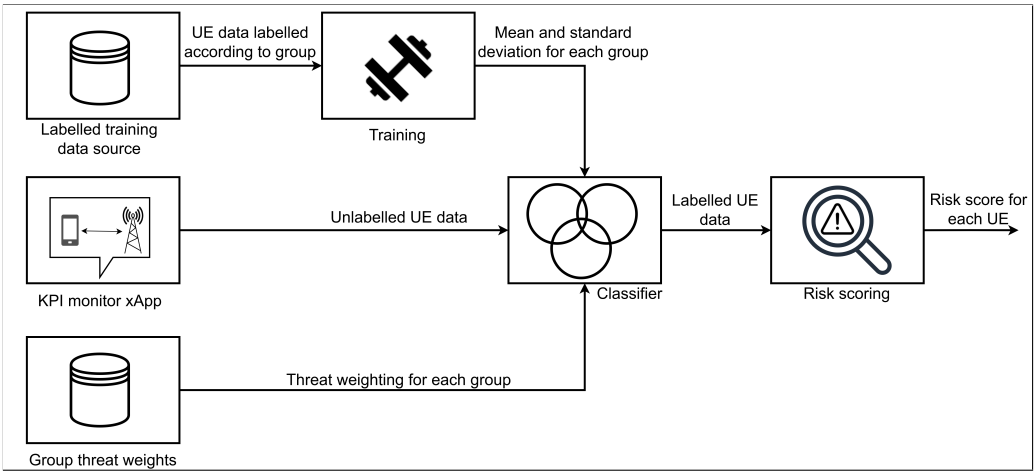

Fig. 2: Risk analysis xApp architecture.

1) Training: Labelled data with one label and n features is input to the training algorithm and scaled to the range $[0, 1]$. Each feature is considered as a dimension. The label of each data point indicates which application group it belongs to. Each group considered has a threat score between 0 and 1 assigned to it. For each label, the mean (μ) and standard deviation (σ) of each feature are found, and the combination of the mean values in each dimension is plotted and considered as the mean for that label. A scaling factor, α, is applied to each standard deviation value calculated. For each group, g, the expected behaviour in each dimension, d is calculated as $\mu_{g,d} \pm \alpha\sigma_{g,d}$, where $\mu_{g,d}$ refers to the mean value of dimension d for group g. These expected behaviour ranges then form the decision boundaries. Any data that falls within a group's expected behaviour range in each dimension is said to be within the absolute behaviour of that group. Care must be taken to ensure that the data used in the training stage is free from data generated by an attacker, otherwise the attacker may be able to manipulate the training stage to their benefit. Attacker-free data could be generated using a private testbed or a simulation.

2) Classification: During the classification step, each unlabelled data point to be classified is input to the algorithm. For each input data point, a list of group memberships is maintained. The data point's group membership list is initially empty. The data point is then compared to the absolute behaviour of each group. If that data point is within the absolute behaviour of a group, then the group's label is added to the data point's group membership list.

3) Risk Scoring: During the risk scoring step, for each data point and group to which the data point is assigned, the Euclidean distance between the data point and the mean of the group is found and scaled by the threat score of the group. For each data point, the largest scaled Euclidean distance is taken as the risk score for that data point. This method could be extended to take into account the Euclidean distance between the data point and labelled attack data, similar to the Cluster Centers algorithm discussed by Castellani et al. [10].

V. Experiment: Risk Analysis Classification

This section details an experiment carried out to demonstrate the operation of the classification step of the risk analysis system as discussed in Sections IV-B1 and IV-B2. The experiment was carried out using a set of Python scripts to process the dataset, as described in the following sections. The dataset used in this experiment is Ofcom's 5G New Radio Mobile Signal Strength Measurement Data for August 2023 [14], which presents data collected regarding the coverage of four mobile network operators (MNOs) in terms of metrics including RSRP, RSRQ, and SINR at locations across the United Kingdom [14]. We use the MNO label in this dataset in place of UE category labels for this experiment. For access to the dataset used in this work, please see section VIII.

A. Data Preprocessing

Prior to any analysis involving the data, the data was subject to two preprocessing steps: averaging by location, and shuffling. During the averaging step, all data points with the same latitude and longitude were averaged. The data for each MNO was then accumulated and labelled. Any rows containing missing values for RSRP, RSRQ, or SINR were dropped. Each row in the resulting comma-separated values (CSV) file contained the RSRP, RSRQ, SINR, MNO label, latitude and longitude for an MNO at a given location. During the shuffling step, the rows of the averaged dataset were randomly swapped to shuffle the data. This counteracted the bias introduced in the averaging step, where the data was ordered by MNO label. The two data preprocessing steps of averaging and shuffling were carried out using the Northern Ireland High-Performance Computing (NI-HPC) service's Kelvin2 HPC cluster due to the large memory requirement of these steps.

B. Data Split and Scaling

The test set size was 25% of the preprocessed dataset, and the training set was 75% of the preprocessed dataset. The RSRP, RSRQ, and SINR were used as the input data, and the MNO labels were used as the data to be predicted. The

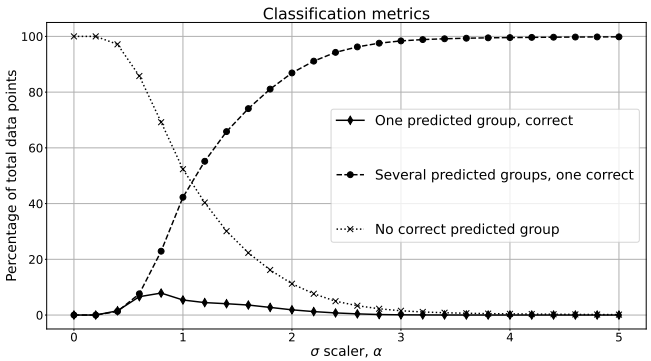

Fig. 3: Classification accuracy metrics.

training input data and the test input data were both scaled independently of each other over the range [0,1].

C. Training

During the training step, for each MNO, the mean and standard deviation of each feature were calculated. With each feature considered as a dimension, the upper and lower classification bounds in each dimension for MNO were calculated as in section IV-B. This resulted in a table of the upper and lower bounds in each dimension (i.e., the absolute behaviour) for each MNO.

D. Classification Testing and α Hyperparameter

During the testing of the classification step, each row in the test data was compared to the absolute behaviour for each MNO. For each test data point, a list of predicted MNOs was maintained. If the data point fell within the absolute behaviour boundaries of an MNO, then that MNO was added to the data point's predicted MNO list. If a data point did not fall within the absolute behaviour of any MNO, then "unclassified" was appended to its predicted MNO list. To understand the relationship between the α hyperparameter, used to scale the classification boundary for each feature, and classification accuracy, a parameter sweep of this variable has been performed. During the parameter sweep, the training and classification steps were repeated for values of α between 0 and 5. The resulting data is presented in Fig. 3, which illustrates the percentage of test data which falls into each of the following categories:

- Only one MNO is predicted, and this prediction is correct
- Several MNOs are predicted, and one is correct
- Correct MNO is not present in predicted MNOs list

From Fig. 3, it may be observed that the selection of the value of α is a balance between minimising the proportion of data points which are not classified under any MNO, and maximising the proportion of data points which have one group correctly assigned. From the data provided in Fig. 3, it might be decided that 2 is a suitable value of α, as the proportion of unclassified data points may be suitably low, while the proportion of data points with one correct MNO assigned converges towards 0 beyond this α value.

VI. Open Research Problems

This section presents further research problems related to this work which have not yet been resolved.

A. Defending Against Malicious xApps

This problem considers an enhanced version of the earlier explained threat model. The attacker now has access to a compromised xApp running on the Near-RT RIC, through which they may compromise other xApps running on the same Near-RT RIC. We propose the development of two further systems, and the combination of these systems with the UE risk analysis system for defence against adversaries with access to compromised xApps.

1) xApp Misbehaviour Detection System: This proposed system analyses network traffic to and from the xApps in the Near-RT RIC in real-time to detect unusual activity. If a sufficiently large deviation from normal operating conditions is observed in the data, the relevant xApp is said to be misbehaving. This misbehaviour detection system will derive a misbehaviour score for each xApp, and if the score is higher than a set value, will alert a security operations centre (SOC) and/or cause the Near-RT RIC platform to take automatic actions such as isolating or terminating the xApp in question.

2) Resilience Management System: If a deployed xApp is found to be misbehaving, then its functionalities must be carried out elsewhere in the network while the misbehaviour is addressed (e.g., through quarantine or redeployment) in order to maintain its availability. We propose the implementation of a resilience management system, which receives information regarding xApp misbehaviour from the xApp misbehaviour detection system, is aware of computational resources available in the connected Near-RT RICs and O-Cloud, and is capable of automatically redeploying misbehaving xApps in another location in the network if required. This resilience management system should implement a zero-touch commissioning methodology, e.g., as discussed by Angui et al. [15], such that xApps may be redeployed without the need for human input. This automatic redeployment of xApps should enhance the survivability of the network against the attacker in the enhanced threat model.

3) Combination of Proposed Solutions: When considering combining the UE risk analysis, xApp misbehaviour detection, and resilience management systems proposed in this work, the locations of these functions within the O-RAN network must be decided. As discussed in section IV, the UE risk analysis system would suit implementation as an xApp. The xApp misbehaviour detection system must be capable of receiving all network traffic to and from xApps on the Near-RT RIC. As our threat model involves an adversary which may attack other xApps on the same Near-RT RIC, a function which is trusted to identify xApps misbehaving should not be an xApp itself. This function could, therefore, be implemented as a security functionality within the Near-RT RIC platform. The resilience management system must receive information from the xApp misbehaviour detection system, receive information regarding the network topology, and redeploy xApps. Due to

TABLE I: Potential O-RAN KPIs for UE risk analysis, xApp misbehaviour detection, and O-RAN resilience management (P1: UE risk analysis, P2: xApp misbehaviour detection, P3: O-RAN resilience management).

Use case / KPI	P1	P2	P3	Entities
RSRP	✓	X	X	UE, O-RU, E2 nodes
RSRQ	✓	X	X	UE, O-RU, E2 nodes
SINR	✓	X	X	UE, O-RU, E2 nodes
Uplink and downlink bitrates	✓	X	X	UE, O-RU, E2 nodes
Uplink and downlink bit error rates	✓	X	X	UE, O-RU, E2 nodes
Request round trip time	X	✓	X	xApp, Near-RT RIC
Packet size	✓	✓	X	UE, xApp, E2 nodes, Near-RT RIC
Destination IP and port	X	✓	X	xApp, Near-RT RIC
Malformed packet rate	✓	✓	X	UE, xApp, E2 nodes, Near-RT RIC
Output from intrusion detection system	X	✓	X	xApp, Near-RT RIC
Output from anomaly detection system	✓	✓	X	UE, xApp, E2 nodes, Near-RT RIC, AD xApp
Number of misbehaving xApps	X	X	✓	xApp misbehaviour detection system, Near-RT RIC
Number of misbehaving virtual/cloud network functions	X	X	✓	SMO, cloud network functions, virtual network functions
Available resources in O-Cloud	X	X	✓	SMO, O-Cloud

the requirement for this function to manage xApp deployments, it is implemented as a rApp in the Non-RT RIC, as rApps may access the O1 and O2 interfaces to manage xApps. The locations of these proposed components within an O-RAN network are illustrated in Fig. 4.

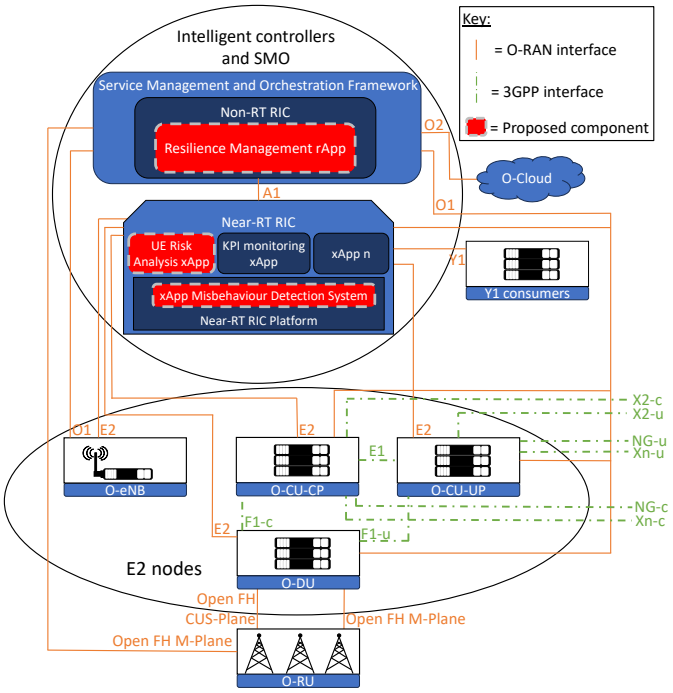

Fig. 4: O-RAN architecture [1] featuring the proposed misbehaviour detection, risk analysis, and resilience management systems.

B. Identification and Evaluation of O-RAN Security and Resilience KPIs

For a KPI-based O-RAN security or resilience system to function effectively, KPIs must be identified, and then the most impactful KPIs for the given application should be used. We have provided examples of KPIs which may prove useful in UE risk analysis, xApp misbehaviour detection, and resilience management systems in Table I.

C. Improvements to the UE Risk Analysis System

Three areas of improvement have been identified to expand upon the UE risk analysis system proposed in this work: the use of UE data in future experiments, the implementation of the risk analysis system as an xApp, and the analytical optimisation of the α hyperparameter. The use of UE data from an O-RAN testbed or simulation will allow for more realistic insights into the performance of the system, and the implementation of the system as an xApp will be critical for its integration into O-RAN networks. The parameter sweep carried out in Section V may not be the most computationally efficient way to determine the optimal value of α. Further work should be carried out to determine a computationally cheaper analytical method of optimising α to obtain desired classification behaviour.

VII. Conclusion

This work has introduced current threat modelling methods related to O-RAN networks, security threats facing O-RAN networks, and technologies supporting the mitigation of these threats. We have presented a threat model of an adversary utilising UEs to poison the data used to train ML models in the Near-RT RIC and Non-RT RIC, and discussed how our risk analysis technique may identify unusual behaviour in

UEs relative to their group. We finally discussed open research problems, including extending our solution to offer resilience against a more capable attacker through monitoring xApp network traffic and zero-touch redeployment of compromised xApps.

VIII. Data and Code Availability

The August 2023 5G NR mobile signal strength measurement dataset used in Section V is available from Ofcom [14]. The code used to carry out the experiment in Section V is available on request to the authors.

Acknowledgements

This work is partly funded by the UK Government through the New Deal for Northern Ireland (NICYBER'25). The funding is delivered on behalf of the Northern Ireland Office and the Department for Science, Innovation and Technology by Innovate UK. Additionally, this work was supported by the UK Department for Science, Innovation and Technology under the Future Open Networks Research Challenge project TUDOR (Towards Ubiquitous 3D Open Resilient Network). We are also grateful for the use of the computing resources from the Northern Ireland High Performance Computing service funded by EPSRC (EP/T022175). The work of B. Canberk is supported in part by The Scientific and Technological Research Council of Turkey (TUBITAK) Frontier R&D Laboratories Support Program for BTS Advanced AI Hub: BTS Autonomous Networks and Data Innovation Lab Project 5239903. The work of T. Q. Duong was supported in part by the Canada Excellence Research Chair (CERC) Program CERC-2022-00109 and in part by the Natural Sciences and Engineering Research Council of Canada (NSERC) Discovery Grant Program RGPIN-2025-04941.

References

[1] O-RAN Alliance, *O-RAN alliance specifications*. [Online]. Available: https://specifications.o-ran.org/specifications (last accessed Jan. 27, 2025).

[2] S. Mahboob, J. Dai, A. Soysal, and L. Liu, "Transforming future 6G networks via O-RAN-empowered NTNs," *IEEE Communications Magazine*, pp. 1–7, 2025. DOI: 10.1109/MCOM.001.2400104.

[3] J. Groen *et al.*, "Securing O-RAN open interfaces," *IEEE Transactions on Mobile Computing*, vol. 23, no. 12, pp. 11 265–11 277, 2024.

[4] E. Habler *et al.*, "Adversarial machine learning threat analysis and remediation in open radio access network (O-RAN)," *Journal of Network and Computer Applications*, vol. 236, p. 104 090, 2025.

[5] J. Groen *et al.*, "Implementing and evaluating security in O-RAN: Interfaces, intelligence, and platforms," *IEEE Network*, vol. 39, no. 1, pp. 227–234, 2025.

[6] O-RAN Alliance Work Group 1, "Use cases analysis report," Technical Report, Jun. 2024 (Last Accessed: October 2024), https://specifications.o-ran.org/specifications.

[7] H. Alimohammadi *et al.*, "KPI poisoning: An attack in Open RAN near real-time control loop," in *Future Networks World Forum*, Dubai, UAE, 2024.

[8] S. Soltani, M. Shojafar, A. Brighente, M. Conti, and R. Tafazolli, "Poisoning bearer context migration in o-ran 5g network," *IEEE Wireless Communications Letters*, vol. 12, no. 3, pp. 401–405, 2023. DOI: 10.1109/LWC.2022.3227676.

[9] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, *NIST special publication 800-207, Zero trust architecture*, Aug. 2020. [Online]. Available: https://doi.org/10.6028/NIST.SP.800-207.

[10] A. Castellani, S. Schmitt, and S. Squartini, "Real-world anomaly detection by using digital twin systems and weakly supervised learning," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 7, pp. 4733–4742, 2021.

[11] HCL Technologies Limited. "Ric-app/ad." (2024), [Online]. Available: https://gerrit.o-ran-sc.org/r/admin/repos/ric-app/ad (last accessed May 1, 2024).

[12] T. Ahmed, A. Shah, M. Kolla, and R. Yellasiri, "Reduction of alert fatigue using extended isolation forest," in *International Conference on Forensics, Analytics, Big Data, Security (FABS)*, vol. 1, 2021, pp. 1–5. DOI: 10.1109/FABS52071.2021.9702617.

[13] Z. Wang, Y. Zhou, H. Liu, J. Qiu, B. Fang, and Z. Tian, "ThreatInsight: Innovating early threat detection through threat-intelligence-driven analysis and attribution," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 12, pp. 9388–9402, 2024.

[14] Ofcom, *Mobile signal strength measurement data*, Mar. 2024. [Online]. Available: https://www.ofcom.org.uk/phones-and-broadband/coverage-and-speeds/mobile-signal-strength-measurement-data/ (last accessed Jan. 22, 2025).

[15] B. Angui, R. Corbel, V. Q. Rodriguez, and E. Stephan, "Towards 6G zero touch networks: The case of automated Cloud-RAN deployments," in *19th Annual Consumer Communications & Networking Conference (CCNC)*, 2022, pp. 1–6.

Mark Megarry received the MEng degree in Electrical and Electronic Engineering from Queen's University Belfast (QUB) in 2023. He is currently pursuing a PhD degree with QUB on the security of 6G O-RAN.

Vishal Sharma is a Reader (∼Senior Associate Professor) in the School of Electronics, Electrical Engineering and Computer Science (EEECS) at Queen's University Belfast (QUB), Belfast, United Kingdom.

Berk Canberk is a Professor in the School of Computing, Engineering and The Built Environment at Edinburg Napier Unuversity, United Kingdom.

Trung Q. Duong is a Canada Excellence Research Chair (CERC) and Full Professor at Memorial University, Canada. He is also an adjunct professor at Queen's University Belfast, UK. He is a Fellow of IEEE, IET, CAE, EIC, and AAIA.