This paper has been accepted for publication on the 1st free5GC World Forum in conjunction with ACM CCS 2025

Robust Anomaly Detection in O-RAN: Leveraging LLMs against Data Manipulation Attacks

Thusitha Dayaratne
Monash University, Australia

Ngoc Duy Pham
Monash University, Australia

Viet Vo
Swinburne University of Technology
Australia

Shangqi Lai
CSIRO's Data61, Australia

Sharif Abuadbba
CSIRO's Data61, Australia

Hajime Suzuki
CSIRO's Data61, Australia

Xingliang Yuan
The University of Melbourne
Australia

Carsten Rudolph
Monash University, Australia

Abstract

The introduction of 5G and the Open Radio Access Network (O-RAN) architecture has enabled more flexible and intelligent network deployments. However, the increased complexity and openness of these architectures also introduce novel security challenges, such as data manipulation attacks on the semi-standardised Shared Data Layer (SDL) within the O-RAN platform through malicious xApps. In particular, malicious xApps can exploit this vulnerability by introducing subtle Unicode-wise alterations (hypoglyphs) into the data that are being used by traditional machine learning (ML)-based anomaly detection methods. These Unicode-wise manipulations can potentially bypass detection and cause failures in anomaly detection systems based on traditional ML, such as AutoEncoders, which are unable to process hypoglyphed data without crashing. We investigate the use of Large Language Models (LLMs) for anomaly detection within the O-RAN architecture to address this challenge. We demonstrate that LLM-based xApps maintain robust operational performance and are capable of processing manipulated messages without crashing. While initial detection accuracy requires further improvements, our results highlight the robustness of LLMs to adversarial attacks such as hypoglyphs in input data. There is potential to use their adaptability through prompt engineering to further improve the accuracy, although this requires further research. Additionally, we show that LLMs achieve low detection latency (under 0.07 seconds), making them suitable for Near-Real-Time (Near-RT) RIC deployments.

1 Introduction

Deployments of 5G networks have significantly enhanced the consumer experience in mobile communications. Its' core capabilities, including enhanced Mobile Broadband (eMBB), massive Machine-Type Communications (mMTC), and Ultra-Reliable Low-Latency Communication (uRLLC), open immense opportunities across numerous sectors. However, the widespread deployment, architectural complexity, and safety-critical nature of these networks make them attractive targets for various adversaries, ranging from individual malicious actors to nation-states.

Conference acronym 'XX, Woodstock, NY
2018. ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

Several vulnerabilities exist in Layer-3 protocols, including the Radio Resource Control (RRC)[1], which manages connection setup and bearer configuration, and the Non-Access Stratum (NAS)[2] that is responsible for mobility and session management between User Equipments (UEs) and core networks. Given the critical nature of these protocols and insufficient integrity protection/encryption in some messages [3], adversaries are attracted to these. Exploiting vulnerabilities of these protocols enables attacks such as Blind Denial of Service (DoS), Null cipher, Downlink DoS, and Lullaby attacks, causing session disruptions, battery drainage, or compromised security setups [4–6]. Despite the security measures defined by the 3GPP, substantial enhancements to these protocols are challenging and often deferred to future standards given the backward compatibility issues. Thus, it is essential to detect these attacks as early as possible. However, detecting these attacks is challenging within traditional monolithic architectures given vendor-specific constraints.

In recent years, the Open Radio Access Network (O-RAN) architecture is emerging as a solution to enhance the controllability, security, and visibility of cellular networks by decoupling RAN hardware and software components and standardising interfaces. This decoupling allows network operators and third-party developers to implement flexible, scalable, and vendor-agnostic monitoring and response solutions. In particular, Near Real time RAN Intelligent Controller (Near-RT RIC) can be leveraged to inspect and control the control-plane messages with the use of Artificial Intelligence and Machine Learning (AI/ML) modules, often implemented as xApps (applications that runs on Near-RT RIC). Some recent studies have already explored the potential for both rule-based [3] and traditional ML-based methods [7, 8] for Layer-3 attack detection within the O-RAN context. Many existing anomaly detection methods deployed within O-RAN rely on shared data layers (SDL) to retrieve appropriate data. However, the SDL interface is not yet fully standardised given the novelty of O-RAN concept. Thus, this under standardised nature of the SDL creates a potential attack surface. In particular, malicious or compromised xApps could potentially manipulate or obfuscate message data. For example, manipulation can be subtle alterations to message names or parameters, such as using "hypoglyphs" (visually similar, distinct Unicode variations) to bypass traditional ML-based models [3, 8] as these models require precise patterns or statistical features for effective detection.

To address these critical vulnerabilities, we propose leveraging Large Language Models (LLMs) for anomaly detection in 5G Layer-3 protocols. In particular, we attempt to leverage LLMs as a more robust approach against data manipulation attacks originating from the SDL. Unlike traditional ML approaches that rely on syntactic patterns, LLMs possess capabilities to identify semantically equivalent inputs (despite syntactically manipulated), which makes LLMs resilient against data manipulation attacks, such as hypoglyphing. Our preliminary analysis depicts that LLMs based anomaly detection methods are resilient against such attacks and can complete the detection within O-RAN time constraints. To the best of our knowledge, this work is the first to specifically investigate the feasibility, robustness, and real-time practicality of LLM-based anomaly detection against data manipulation attacks within the SDL context of 5G O-RAN environments.

2 System Overview

2.1 Threat Model

While numerous attack vectors exist in the O-RAN context, including fake O-DU/O-RU nodes, fake UEs, and jammers, this work focuses on a highly plausible and sophisticated threat of malicious xApps deployed on the Near-RT RIC. Unlike more resource-intensive attacks (e.g., enrolling fake E2 nodes or jamming), deploying malicious xApps requires minimal financial investment, which makes it a significant threat.

Attackers can exploit the under standardised and evolving nature of the xApp platform to deploy such malicious xApps on the Near-RT RIC. Security vulnerabilities within the platform and lack of in-depth evaluation procedures could enable these deployments. Additionally, adversaries can exploit zero-day vulnerabilities similar to the other common well-established application platforms such as iOS and Android app stores despite existing security measures. We assume the malicious xApp possesses read/write access to the SDL. This assumption aligns with recent findings [9, 10] and observations from the O-RAN Working Group 11 study on Near-RT RIC and xApp security [11], which denote the absence of robust authentication or authorisation for databases, including the SDL, within the O-RAN ecosystem.

Our threat model specifically considers a malicious xApp exploiting its SDL access to manipulate data that is consumed by a xApps, which uses that data for anomaly detection based on traditional ML model. The adversary's objective is to evade detection by these existing ML models. However, instead of deleting on direct corruption of the data, we assume the malicious xApp introduces subtle Unicode-wise alteration ("hypoglyphs"—characters that are visually similar) to legitimate Layer-3 messages. For example, the malicious xApp can modify *RRCSetupRequest* message to *RRCSetupRequest* where despite looks exactly same, here Latin "C" (U+0043) was replaced with a Cyrillic "C" (U+0421), Latin "e" (U+0065) with a Cyrillic "e" (U+0435) and Latin "q" (U+0071) with a visually similar character (U+055B, Armenian small letter q).

2.2 Framework

A high-level overview of the LLM-based detection framework is depicted in Figure 1. The framework is deployed as an xApp within the Near-RT RIC and leverages the RRC and NAS messages, along with associated TMSI and RNTI identifiers that are stored in the SDL [3].

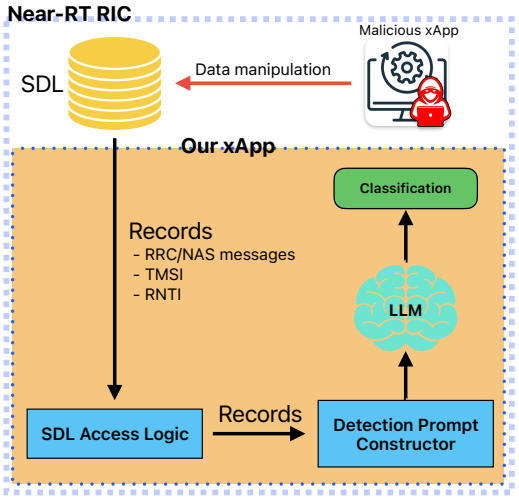

Figure 1: High-level system overview

The SDL Access Logic module periodically queries the database to retrieve the latest RRC/NAS messages for attack detection. Upon retrieval, the messages are sent to Detection Prompt Constructor module. This module integrates the messages and a description about the task to construct the detection prompt, which is then use with the LLM to perform the classification. The LLM provides the classification (Normal or Anomalous).

3 Implementation

3.1 Dataset

We utilise the dataset from previous work [8], which comprises data collected from a physical testbed involving four distinct 5G smartphone models. This dataset also incorporates data generated using the COLOSSEUM [12] wireless network emulator. The dataset was structured in an ordered fashion, where it group messages sequentially by each UE (e.g., $msg_1^{UE_1}, msg_2^{UE_1}, \ldots, msg_1^{UE_2}, msg_2^{UE_2}, \ldots, msg_1^{UE_3}, msg_2^{UE_3}, \ldots$). However, the original dataset contains only three instances of Blind DoS attacks. Thus, we augmented the dataset by introducing 17 additional Blind DoS attack instances, which resulted about 1% overall attack instances. In particular, we injected 17 *RRCSetupRequest* messages with preselected existing TMSI values and random RNTI values. The enhanced dataset consists of 1,016 RRC and NAS messages, including 20 Blind DoS attack scenarios.

Additionally, we further manipulated 5 messages within this enhanced dataset using Unicode-wise alteration, as discussed in our threat model in order to evaluate the resilience against evasion attacks. These manipulated messages include 2 of the Blind DoS attack instances and 3 normal messages. In particular, we changed some of the characters in those messages using alternative characters.

We then adopted a window-based approach with overlapping windows. Specifically, we empirically evaluated window sizes ranging from 1 to 10. A window size of 1 corresponds to the non-overlapping scenario, where each message is processed independently along with its corresponding previous message. In contrast, a window size of 10 indicates that each new incoming message is combined with the nine preceding messages. The corresponding previous message for the latest message is also included in the prompt. At any given time, we used only one new incoming message to enable streaming-based detection. This approach mitigates the inherent latency of batch processing, which occurs when multiple new messages are processed together. Such delays can allow malicious messages to remain undetected for extended periods and may require deferring processing until a predefined message threshold is reached.

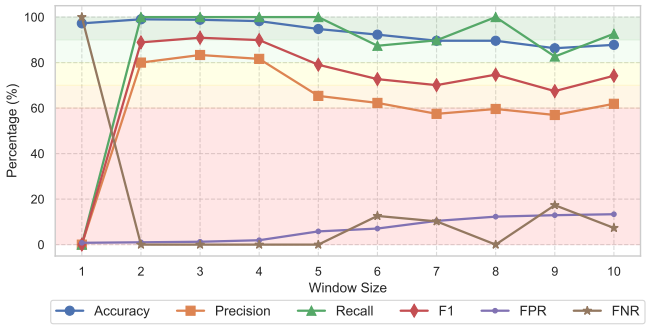

Figure 2: Detection performance of AutoEncoder model under normal scenario

3.2 LLM

We used the Meta's Llama-3.1-8B-Instruct model. The open-source model was obtained from HuggingFace[13] and deployed using vLLM[14] on an NVIDIA A100 GPU (80GB). We did not pre-train or fine-tune the model. The temperature parameter was set to 0 to ensure deterministic outputs and prevent LLMs from hallucinations or creative deviations in the predictions.

4 Evaluation & Results

To evaluate the performance of the proposed framework, we aim to answer the following two research questions.

RQ-1 How does the presence of Unicode-wise manipulated messages (hypoglyphs) in the SDL impact the detection capabilities of traditional ML-based anomaly detection xApps?

RQ-2 Does an LLM-based anomaly detection xApp demonstrate robustness against Unicode-wise manipulated (hypoglyphed) messages, and can its detection performance be effectively improved through prompt engineering?

We primarily use Accuracy, Precision, Recall, F1 score, False Positive Rate (FPR), and False Negative Rate (FNR) for the evaluation. However, we note that achieving a high F1 Score along with low FPR and FNR is the desired outcome.

4.1 RQ-1 Unicode-wise manipulated Layer-3 messages with traditional ML model

Under this research question, we analysed the detection capabilities and robustness of existing traditional ML-based anomaly detection xApps when messages contain Unicode-wise manipulations (hypoglyphs). In particular, we aim to determine if these xApps can handle such adversarial attacks.

We utilised the AutoEncoder model and the code from the 6G-XSec [8], an existing traditional ML-based anomaly detection xApp designed to identify Layer-3 attacks for this evaluation. We trained 10 distinct AutoEncoder models, where each model corresponding to a different window size ranging from 1 to 10. For each model, the first 700 normal messages from the dataset (with attacks filtered out) were used for training. The remaining data was used for testing.

Figure 2 depicts the detection performance of the models when tested without any hypoglyphed messages, where the F1 value remain over 75% with most of the window sizes.

Then we tested the same models using the manipulated dataset, which includes 5 hypoglyphed messages. All 10 traditional ML-based AutoEncoder models experienced an immediate failure/crash upon the encounter of the first hypoglyphed message in the test set. The failure prevented any further anomaly detection. This failure is directly related to the AutoEncoder's underlying feature extraction and encoding mechanisms. In particular, the models were trained using a fixed set of normal, well-formed message patterns. Thus, they were unable to handle the novel, unseen Unicode characters introduced by the hypoglyphs. Unlike a misclassification, which would yield an incorrect output, the models' core processing pipelines failed. We denote that error handling could be implemented to prevent a full crash. However, that could potentially result in the model skipping or failing to process the un-encodable messages, which hinders the effectiveness of traditional ML-based anomaly detection against these evasion tactics.

4.2 RQ-2 Effectiveness of LLM based Detection

Under this research question, we primarily evaluated the robustness of an LLM-based anomaly detection xApp against hypoglyphed messages. Additionally, we also seek to analyse whether detection performance can be improved through prompt engineering. For this evaluation, we implemented an LLM-based anomaly detection xApp as discussed in section 2.2. The LLM-based xApp was evaluated using the complete dataset of 1,016 messages, which included the 5 hypoglyphed messages. Unlike in the AutoEncoder model, we initialised the same LLM model 10 times, corresponding to window sizes ranging from 1 to 10.

Unlike the traditional ML-based AutoEncoder models, which experienced failures upon encountering hypoglyphed messages, LLM-based models showed the needed robustness. In particular, the LLM-based models successfully processed every single message in the dataset, including all Unicode-wise manipulated instances, without any system crashes or early terminations. This behaviour highlights that LLMs are not susceptible to the encoding and feature extraction failures, which hinder the impact of traditional models against unseen or subtly altered characters.

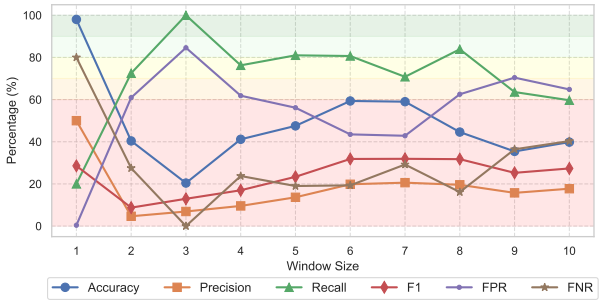

Figure 3: Detection performance of LLM-based models with hypoglyphed messages

Table 1: Average detection time per message for LLM

Window Size	Average Time (seconds)
1	0.030
2	0.046
3	0.056
4	0.052
5	0.054
6	0.056
7	0.056
8	0.067
9	0.069
10	0.070

Figure 3 depicts the detection performance of the LLM-based xApp across various window sizes. The F1-scores ranged from about 0.087 to 0.319 across different window sizes. Despite the fact that the LLM successfully processed all inputs without crashing, these initial F1-scores suggest that the out-of-the-box anomaly detection capabilities for these Layer-3 attacks require further optimisation and prompt engineering.

Table 1 shows the average detection time per message for each window size with the LLM-based approach. The average detection times ranged from 0.03 seconds for a window size of 1 to 0.07 seconds for a window size of 10. It is clear that the LLM requires more time when the number of messages in the window increases. However, these times are well within the Near-RT RIC's real-time constraint of 1 second, which confirms the practical feasibility of LLM-based detection in terms of latency.

5 Discussion & Conclusion

Our evaluation shows a critical issue in the pipeline of traditional ML-based anomaly detection methods in the O-RAN context, when exposed to semantically altered inputs, such as hypoglyphs. Instead of simple misclassifications, these inputs lead the models to operational failure upon processing manipulated data. This vulnerability is highly critical in the context of the O-RAN architecture, where a malicious xApp can exploit the evolving standardisation of the SDL to disable security monitoring systems without triggering conventional detection mechanisms. Such failures can potentially result in total loss of detection capabilities and leading the network vulnerable additional compromises.

In contrast, our LLM-based anomaly detection shows resiliency against manipulated (hypoglyphed) messages. This resilient nature of LLMs highlights LLM-based solutions as a reliable alternative to conventional ML models for security-critical deployments. Even though baseline detection performance requires additional optimisation, our findings show that LLMs can be used to implement anomaly detection solutions that are more robust against crashes from manipulated input data. We suggest that given the appropriate prompt engineering, their performance in terms of accuracy could be further improved without retraining. Additionally, our detection achieves below 0.1-second detection latency, which demonstrates its suitability for deployment in Near-RT RIC environments. These results demonstrate the potential of LLMs to provide robust, low-latency detection of Layer-3 attacks, even with advanced data manipulations within the SDL, contributing to the development of more secure and resilient O-RAN networks.

Acknowledgement

This research paper is conducted under the 6G Security Research and Development Project, as led by the Commonwealth Scientific and Industrial Research Organisation (CSIRO) through funding appropriated by the Australian Government's Department of Home Affairs. This paper does not reflect any Australian Government policy position. For more information regarding this Project, please refer to https://research.csiro.au/6gsecurity/.

References

[1] 3GPP. Ts 38.331: Nr; radio resource control (rrc); protocol specification. Technical Report Release 15, 3rd Generation Partnership Project (3GPP), 2018. URL https://www.3gpp.org/ftp/specs/archive/38_series/38.331/.

[2] 3GPP. Ts 24.501: Non-access-stratum (nas) protocol for 5g system (5gs); stage 3. Technical Report Release 15, 3rd Generation Partnership Project (3GPP), 2018. URL https://www.3gpp.org/ftp/Specs/archive/24_series/24.501/.

[3] Haohuang Wen, Phillip Porras, Vinod Yegneswaran, Ashish Gehani, and Zhiqiang Lin. 5g-spector: An o-ran compliant layer-3 cellular attack detection service. In *Proceedings of the 31st Annual Network and Distributed System Security Symposium, NDSS*, volume 24, 2024.

[4] Syed Rafiul Hussain, Mitziu Echeverria, Imtiaz Karim, Omar Chowdhury, and Elisa Bertino. 5greasoner: A property-directed security and privacy analysis framework for 5g cellular network protocol. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*, pages 669–684, 2019.

[5] Hongil Kim, Jiho Lee, Eunkyu Lee, and Yongdae Kim. Touching the untouchables: Dynamic security analysis of the lte control plane. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 1153–1168. IEEE, 2019.

[6] Simon Erni, Martin Kotuliak, Patrick Leu, Marc Roeschlin, and Srdjan Capkun. Adaptover: adaptive overshadowing attacks in cellular networks. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, pages 743–755, 2022.

[7] Alessio Scalingi, Salvatore D'Oro, Francesco Restuccia, Tommaso Melodia, and Domenico Giustiniano. Det-ran: Data-driven cross-layer real-time attack detection in 5g open rans. In *IEEE INFOCOM 2024-IEEE Conference on Computer Communications*, pages 41–50. IEEE, 2024.

[8] Haohuang Wen, Prakhar Sharma, Vinod Yegneswaran, Phillip Porras, Ashish Gehani, and Zhiqiang Lin. 6g-xsec: Explainable edge security for emerging openran architectures. In *Proceedings of the 23rd ACM Workshop on Hot Topics in Networks*, pages 77–85, 2024.

[9] Azuka Chiejina, Brian Kim, Kaushik Chowhdury, and Vijay K Shah. System-level analysis of adversarial attacks and defenses on intelligence in o-ran based cellular networks. In *Proceedings of the 17th ACM Conference on Security and Privacy in Wireless and Mobile Networks*, 2024.

[10] Cheng-Feng Hung, You-Run Chen, CHI-Heng Tseng, and Shin-Ming Cheng. Security threats to xapps access control and e2 interface in o-ran. *IEEE Open Journal of the Communications Society*, 2024.

[11] O-RAN ALLIANCE, Work Group 11 (Security Work Group). Study on security for near real time ric and xapps. Technical Report O-RAN.WG11.Security-RIC-xApps-v05.00, O-RAN ALLIANCE, 2024. URL https://specifications.o-ran.org/download?id=626. Accessed: 2025-06-05.

[12] Leonardo Bonati, Pedram Johari, Michele Polese, Salvatore D'Oro, Subhramoy Mohanti, Miead Tehrani-Moayyed, Davide Villa, Shweta Shrivastava, Chinenye Tassie, Kurt Yoder, et al. Colosseum: Large-scale wireless experimentation through hardware-in-the-loop network emulation. In *2021 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, pages 105–113. IEEE, 2021.

[13] Hugging Face. Hugging face, 2025. URL https://huggingface.co/. Accessed: 2025-04-10.

[14] vLLM Project. *vLLM Documentation*, 2025. URL https://docs.vllm.ai/en/latest/. Accessed: 2025-04-10.