FALCON++: A Semi-Supervised Anomaly Detection and Localization Framework for Resilient O-RAN Deployments

Yaswanth Kumar LS, Somya Jain, Michael Suguna Kumar Victor, Abdulla Ovais,
Suresh Kumar Amalapuram, Bheemarjuna Reddy Tamma, and Siva Ram Murthy C
Indian Institute of Technology Hyderabad, India
Email: {cs24resch11007, cs23mtech12011, cs24mtech12010, cs24mtech12014, cs25pdf04, tbr}@iith.ac.in
murthy@cse.iith.ac.in

***Abstract*—Fault detection is a critical problem in Open Radio Access Network (O-RAN) observability, but the availability of labeled fault data is extremely limited, and enumerating all possible fault scenarios is costly and often infeasible. Therefore, it is essential to rely on normal network behavior to enable effective fault detection. In this work, we propose FALCON++, a tailored semi-supervised anomaly detection and localization framework. Specifically, we use a specialized contextual autoencoder that fuses feature-wise gating and hierarchical temporal encoding for the effective reconstruction of normal network behavior. Furthermore, we combine a causal decoder with residual skip connections to improve robustness to data deemed anomalous due to their higher reconstruction error. FALCON++ also addresses scalability issues by shifting from a monolithic system model to CU-DU pairwise inferencing. Through experimental results on synthetic faults generated using STRESS-NG and TC, we show that FALCON++ achieves a high-precision and high-recall operating point with an F1-score of 0.77, thereby enabling anomaly detection and localization.**

I. Introduction

Beyond disaggregation, the cloud-native realization of O-RAN enables a broader convergence of communication and computation. In particular, the AI-RAN Alliance [1] promotes execution of AI workloads on shared Radio Access Network (RAN) infrastructure. This convergence aims to improve infrastructure utilization, and enable advanced AI-driven services at the network edge. However, co-locating AI workloads with RAN functions fundamentally alters the resource dynamics of the underlying system. Compute, memory, and accelerator resources such as GPUs or specialized AI accelerators are no longer dedicated solely to radio processing and may become oversubscribed if not carefully orchestrated [2].

In such shared environments, improper resource allocation logic, suboptimal orchestration policies, or misconfiguration of virtualization layers can lead to severe resource contention. More critically, the impact of these issues is often disproportionate and cascading, where minor deviations in resource availability may produce uneven and compounding effects which progressively propagate across O-RAN components and rapidly escalate into service-affecting faults or failures [3].

Consequently, detecting and localizing such cascading faults requires a centralized framework that provides a holistic view of the O-RAN system across the Open Centralized Unit (O-CU), Open Distributed Unit (O-DU), Open Radio Unit (O-RU), and the underlying cloud infrastructure. Such system-wide visibility is crucial, as faults can propagate across interconnected components, complicating accurate Root Cause Analysis (RCA). Furthermore, effective fault localization in such disaggregated RAN environments requires visibility across multiple layers, as faults may originate from container-level, host-level, or RAN-level resource contention. This need is further illustrated through an example in Section IV-C. Given the scarcity of labeled fault data, we model fault detection as an anomaly detection problem, where faults are treated as deviations from normal network behavior.

Designing an effective anomaly detection system for O-RAN deployments presents several challenges. First, supervised anomaly detection approaches are ineffective due to the lack of access to labeled high-quality anomalous data. Anomalous events in operational networks are rare, diverse, making the collection of representative labeled datasets extremely difficult. Second, O-RAN deployments are highly dynamic, with evolving topology, scale, and workload composition driven by network expansion, optimization, or reconfiguration. In such settings, anomaly detection models that require retraining whenever the topology changes incur significant computational and operational overhead, limiting their practical applicability.

To address these challenges, we propose FALCON++, a centralized, lightweight and scalable anomaly detection and localization framework tailored for cloud-native O-RAN. FALCON++ leverages multi-level telemetry collected at the container, host, and RAN application layers to enable accurate fault localization across the cloud-native stack. It employs an LSTM-based autoencoder–decoder architecture to learn normal behaviour in multivariate O-RAN telemetry data, enabling the identification of deviations indicative of anomalous behavior. Also, scalability is achieved through a CU–DU pairwise detection strategy, which decomposes the global detection problem into manageable units while preserving cross-component correlations. This design enables FALCON++ to scale across diverse O-RAN topologies without requiring extensive retraining.

II. Related Work

Anomaly detection and observability in 5G/O-RAN networks have been widely studied, spanning diverse datasets, machine learning techniques, and O-RAN-specific monitoring frameworks. Existing datasets primarily capture radio- and user-level behavior. For instance, [4] provides a comprehensive 5G dataset with throughput, channel, and contextual metrics from real deployments, while [5] offers large-scale simulation traces. However, these datasets are limited to air-interface measurements and do not expose internal O-CU/O-DU states or cloud infrastructure telemetry required for fault localization in cloud-native O-RAN systems, nor do they capture container-level resource dynamics in disaggregated RAN architectures.

This limited observability complicates the identification and labeling of anomalous behavior, as failures may manifest themselves indirectly across multiple layers of the system. Consequently, anomaly detection in RAN systems has often relied on unsupervised and semi-supervised methods such as autoencoders [6], [7]. Although effective in learning normal behavior, these approaches do not explicitly capture the temporal and cross-layer dependencies inherent in O-RAN environments. More recent work applies ML-driven anomaly detection to O-RAN using standardized KPIs [8]. Although promising, these approaches remain largely KPI-centric and detect anomalies in isolation, without correlating signals across host, container, and RAN application layers, limiting their ability to capture interacting failure modes introduced by RAN disaggregation, virtualization, and cloud-native.

Recent O-RAN-specific efforts have begun to address these limitations by emphasizing observability and explainability. SpotLight [9] proposes an accurate and explainable anomaly detection framework, but it does not explicitly address scalability with increasing numbers of Central Unit (CU)–Distributed Unit (DU) pairs or dynamic topology changes. Procedural learning–based approaches [10] improve robustness and automation for 5G RAN anomaly detection, yet they do not consider semi-supervised learning or scalable decomposition strategies for large-scale O-RAN deployments. In an earlier work [11], we introduced fault prediction in O-RAN using multi-level telemetry which focuses on pro-active fault classification using supervised learning techniques.

In contrast, **FALCON++** leverages multi-level telemetry spanning across host, container and RAN, and adopts a semi-supervised learning paradigm suited to operational settings with limited labeled data for anomaly detection and localization. Most importantly, it introduces a CU–DU pairwise anomaly detection and localization strategy, enabling fine-grained identification of anomalies while scaling across O-RAN deployment scenarios without requiring retraining.

III. Proposed FALCON++ Framework

The design of FALCON++ is motivated by the following objectives: (i) effectively modeling of the complex temporal dependencies and multi-level architectural relationships inherent in O-RAN deployments, (ii) minimizing False Negatives (FN) and False Positives (FP) as anomalies are rare (approx. 2%), (iii) maintaining a computationally lightweight footprint for fault detection models, (iv) ensuring a topology-agnostic, scalable solution, and (v) enabling anomaly localization to identify the specific O-RAN component for faster recovery.

To realize these objectives, FALCON++ introduces a pairwise CU-DU inferencing methodology coupled with an Expert Context Autoencoder (AE) model.

A. Pairwise CU-DU Topology and Feature Engineering

A key contribution of the proposed FALCON++ framework is the **Pairwise CU-DU Inferencing** methodology. Unlike traditional monitoring approaches [9], [11] that treat the network as a monolithic entity, our framework reshapes raw wide-format telemetry into a pairwise long-format. This structural shift is designed to isolate component dependencies and is characterized by the following strategic advantages:

- **Isolation via Pairwise Division**: Monitoring KPIs from an entire O-RAN deployment is typically performed through centralized collection of telemetry from all network components. However, analyzing the complete dataset directly introduces a "high-dimensionality curse". To mitigate this, the collected data is subsequently partitioned into individual CU–DU pairs. This pairwise isolation enables the framework to focus on the specific handshake behaviors and interface performance metrics of each logical link, allowing more precise inference and fault identification. Since the telemetry is first gathered centrally and then logically separated into CU–DU pairs, this approach remains deployment-agnostic and can be applied across diverse O-RAN deployment scenarios.
- **Global-Local Contextual Averaging**: To prevent the Expert Context AE from becoming overly localized, we incorporate a *global–local perspective* by augmenting each instance with (i) mean metrics aggregated from other O-DUs and (ii) the count of sibling O-DUs connected to the same O-CU, collectively referred to as *Other O-DU Context*. This enables the framework to be aware of the local CU-DU pair and also the other influencing O-DUs.
- **Topology-Agnostic Scalability**: Traditional monitoring approaches are topology-dependent and do not generalize well across varying network configurations. A key limitation is that they are trained on a fixed set of features tied to a specific topology (e.g., a fixed number of O-CUs and O-DUs). When the topology becomes dynamic, such as with the addition or removal of components, the feature set and its structure change, causing a mismatch with the expected input dimensionality and necessitating retraining to adapt to the new feature space. In contrast, our pairwise inferencing strategy in FALCON++ avoids this issue by operating on component-level pairs rather than a fixed global feature vector, making it inherently topology-agnostic and enabling it to maintain efficiency and accuracy regardless of changes in O-CU count, O-DU count, or O-DUs per O-CU ratios, without requiring retraining.

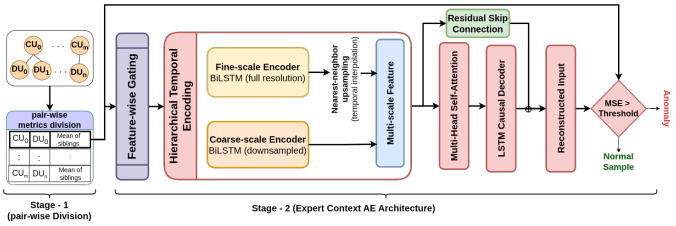

Fig. 1: Proposed FALCON++ Framework.

- **Precision in Anomaly Localization**: Inference at the CU–DU pair granularity inherently enables anomaly localization, where higher reconstruction errors are directly mapped to specific logical links (e.g., `srscu1_srsdu2`), reducing the need for network-wide search for troubleshooting.

Our initial analysis using t-SNE visualization indicates that O-RAN telemetry distributions exhibit significant non-linear structure and class overlap, making it challenging for traditional clustering or density-based methods such as Bayesian Gaussian Mixture Models (BGMM) to establish well-defined decision boundaries for sparse anomalies. Furthermore, we evaluated generative approaches such as Joint-VAEGAN (JV-GAN). While expressive, these models tend to overgeneralize the notion of normality in high-dimensional spaces, resulting in higher false negative (FN) rates, as subtle faults are reconstructed with high fidelity. These limitations motivate the need for a specialized framework that enforces stricter temporal causality and preserves fine-grained signal variations.

To overcome these challenges, we transition from global generative modeling to a context-aware reconstruction task that prioritizes temporal predictive integrity. By integrating multi-scale feature awareness and structural constraints, the proposed framework ensures that deviations from normal network behaviour are statistically amplified rather than smoothed over, directly addressing the requirement for low FN and FP rates in highly dynamic O-RAN environments.

B. Expert Context Autoencoder (Context-AE)

FALCON++ uses an Expert LSTM Autoencoder, specifically engineered for semi-supervised anomaly detection, as illustrated *Stage-2* in Fig. 1. The term "Expert" denotes the explicit incorporation of domain-informed inductive biases that reflect temporal, causal, and resource-driven dynamics of O-RAN systems. We incorporate five specialized enhancements:

1) **Feature-wise Gating**: O-RAN telemetry includes hundreds of noisy or redundant metrics. We utilize learnable feature gates to adaptively weigh inputs. This *suppresses* noise and *amplifies* critical performance indicators (e.g., bitrates, CPU utilization), ensuring the Expert Context AE focuses only on signals that define the network's state.
2) **Hierarchical Temporal Encoding**: Faults manifest at different time granularities, from instantaneous spikes to slow changes that take some time to impact performance of the networks. Modeling choices such as single-resolution LSTM often misses patterns that fall outside its fixed temporal window. We solve this by encoding the telemetry at two different scales.
   - **Fine-Scale Encoding**: Processes full-resolution sequences to capture fast transients.
   - **Coarse-Scale Encoding**: Processes downsampled data to capture global, slow-moving dynamics.

   Combining these scales provides a multi-resolution understanding of network's health.
3) **Multi-Head Self-Attention**: During the transition from encoding to decoding, the model must decide which specific time steps within a window are most relevant to the current network state and give priority accordingly. Certain moments in a sequence are more descriptive of the network's health than others. Self-attention allows the model to weigh the importance of different time steps dynamically. By using multiple "heads", the model can simultaneously focus on different types of temporal relationships, creating a richer representation for the decoder to work with.
4) **Causal Unidirectional Decoder**: The network state at any time step is governed by its historical context. A unidirectional LSTM ensures that reconstruction relies solely on past information, enforcing temporal causality and capturing the progression of normal traffic patterns.
5) **Residual Skip Connections**: Encoder–decoder architectures are prone to reconstruction smoothing, where latent compression attenuates high-frequency temporal variations. To mitigate this, we incorporate residual skip connections that directly propagate encoder feature maps to the decoder. This facilitates the preservation of fine-grained temporal structure, improving reconstruction fidelity for normal patterns. Consequently, anomalous deviations that cannot be effectively reconstructed through these pathways result in amplified reconstruction errors, enhancing detection sensitivity.

C. Anomaly Detection and Localization Logic

The proposed framework identifies anomalies by computing the Reconstruction Error between the input sequence X and its reconstruction $\hat{X}$.

1) **Reconstruction Error (MSE)**: The anomaly score is the Mean Squared Error (MSE) over sequence length w and n features:

$$MSE = \frac{1}{w \cdot n} \sum_{i=1}^{w} \sum_{j=1}^{n} (X_{i,j} - \hat{X}_{i,j})^2$$

2) **Thresholding via Validation**: To minimize false positives in production, we determine the threshold τ empirically by evaluating a range of candidate thresholds on a validation set containing only normal data. Specifically, we analyze the error distribution and select the threshold that provides the best trade-off between sensitivity and false positive rate. In our experiments, the 99th percentile emerged as the most effective operating point. An anomaly is flagged if $Score(X_{pair}) > \tau$.

3) **Computational Efficiency and Localization**: The lightweight LSTM structure enables the framework to operate efficiently. Because the inferencing is pairwise, if `srscu1_srsdu2` reports an MSE exceeding τ, the system immediately identifies the location of the fault, reducing Mean Time to Repair (MTTR).

IV. Experimental Setup

The experimental setup, realized using the *srsRAN* stack in Docker, consists of two O-CUs and three O-DUs. Specifically, O-CU$_0$ serves O-DU$_0$ and O-DU$_1$, while O-CU$_1$ serves O-DU$_2$. Each O-DU is associated with a single User Equipment (UE), enabling precise control over traffic generation. This setup, illustrated in Fig. 2, allows selective stress injection into individual O-RAN components while observing its cascading impact across various levels using telemetry collection.

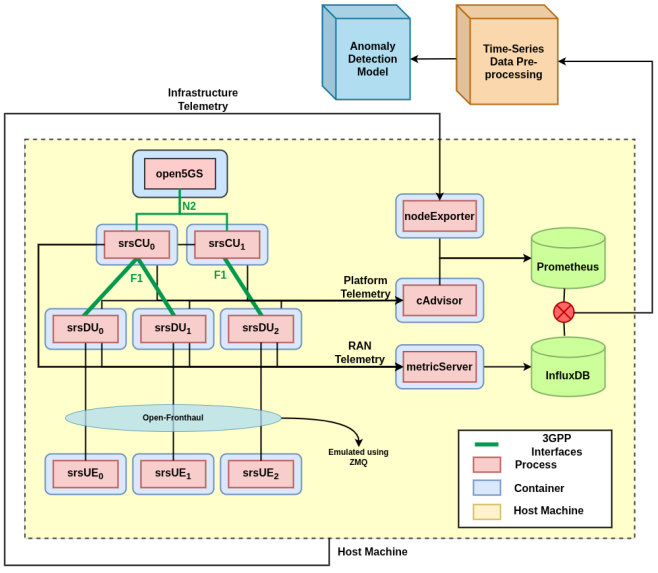

Fig. 2: Experimental O-RAN testbed.

Comprehensive telemetry is collected from host, container, and RAN levels using *node-exporter* [12], *cadvisor* [13], and *srsran metric server* every 1s as described in [11], yielding 403 features across three observability tiers: container-level (7 KPIs for each O-DU and O-CU container - CPU, memory, network I/O, filesystem I/O related KPIs), host-level (271 KPIs - CPU, memory, network/transport, disk, process scheduling, hardware sensors related metrics), and RAN-level (18 KPIs per cell across 3 cells (PCI 1-3, one per O-DU), including UL/DL bitrate, CQI, MCS, HARQ ACK/NACK counts, and PUSCH/PUCCH SNR values).

A. Traffic Generation Model

Uplink traffic is generated from each UE towards the core network using `iperf` in UDP mode. The aggregate offered uplink rate follows a time-varying load profile derived from a canonical 24-hour traffic pattern, as shown in Fig. 3. The total target rate for each hourly interval is divided equally among all active UEs, i.e. $r_u(t) = R(t)/|\mathcal{U}|$, where $R(t)$ is the aggregate offered load at time t. Each UE transmits continuously at its assigned rate within each interval. The 24-hour traffic profile is time-compressed by a factor of four, such that each one-hour segment is executed over a 15-minute interval, resulting in a six-hour traffic trace that preserves the relative temporal load variations of the original pattern.

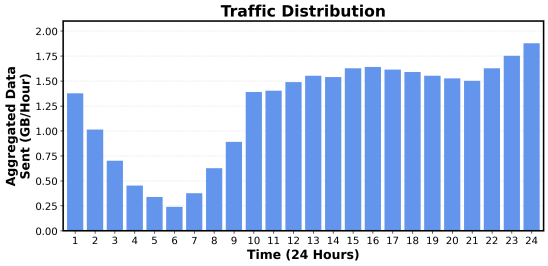

Fig. 3: Traffic pattern.

Unlike TCP-based traffic generation using `iperf`, for UDP-based flows, the offered load $r_u(t)$ remains invariant under faulty network conditions. Resource contention or packet loss occurring at the container, host, or network layer therefore does not reduce the application's sending rate, but instead propagates through lower layers of the RAN protocol stack. This leads to indirect manifestations such as increased buffer occupancy, elevated retransmissions, scheduling delays, and distorted bitrate measurements at the O-DU. Consequently, anomalies can be reliably detected through cross-layer and multi-level telemetry analysis as shown in Fig. 4, motivating the design of the proposed FALCON++ framework.

B. Fault Injection Model

To evaluate the robustness and localization capability of FALCON++ under realistic cloud-native O-RAN deployments, we employ an automated fault injection framework that introduces controlled resource and network impairments directly at the container level. The injected faults emulate operational issues arising from misconfigurations, resource overcommitment, and contention caused by co-located AI and RAN workloads in shared cloud environments.

We consider three primary fault categories, each targeting a distinct dimension of the cloud-native RAN stack:

- **CPU contention**: Emulates compute saturation caused by excessive background workloads, leading to delayed scheduling and degraded processing performance in O-CU and O-DU containers.
- **Memory pressure**: Models memory contention due to memory-intensive workloads within constrained container limits, resulting in paging, buffer buildup, and reduced processing efficiency.
- **Packet loss**: Represents network impairments such as congestion or partial link degradation, injected at container network interfaces to emulate realistic packet loss in virtualized cloud networks.

CPU and memory stress are injected using `stress-ng`, while packet loss is introduced using Linux traffic control

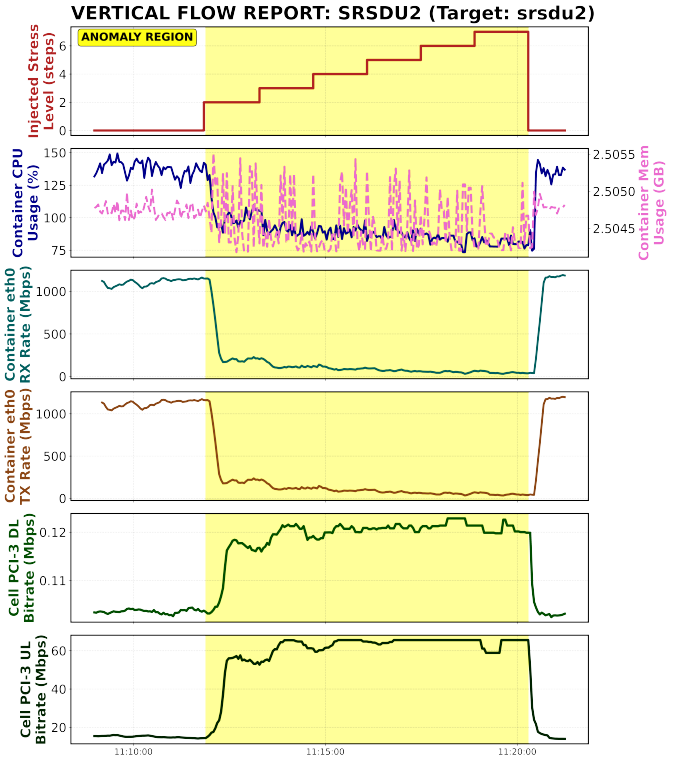

Fig. 4: Network stress testing for UDP traffic.

(`tc`) with `netem` rules applied at the container network interface. Faults are injected independently across O-CU and O-DU containers, with explicit cleanup before and after each injection to prevent residual effects.

Fault injection is performed in a randomized yet controlled manner. At each iteration, a fault type and a target O-CU or O-DU container are randomly selected, and the fault is applied for a duration uniformly sampled between 45–90 s. Stress intensity is calibrated to induce measurable performance degradation without causing container crashes or system instability. A proportional cool down interval is enforced after each fault to maintain an average anomaly density of 10% (approx.).

Randomization is used to reflect the inherently unpredictable nature of faults in operational cloud-native O-RAN deployments, where performance degradation can arise from fluctuating AI workloads, transient misconfigurations, and non-deterministic scheduling behavior. By avoiding deterministic fault patterns, the injection strategy generates diverse and unbiased failure scenarios, preventing overfitting to specific fault signatures and enabling a realistic evaluation of cross-layer anomaly detection and fault localization in FALCON++.

C. Illustrative Example: Packet Loss and Cross-Layer Effects

Fig. 4 illustrates the cross-layer impact of network contention on UDP uplink traffic. Since UDP lacks congestion control and reliable delivery, the UE maintains its transmission rate despite packet loss, triggering aggressive Hybrid-ARD (HARQ) retransmissions at the MAC layer in 5G. This retransmission overhead inflates buffer occupancy and radio-level throughput metrics of O-DU, creating a false impression of high traffic demand. But the application throughput drops due to network congestion. The CPU usage also falls because packet loss prevents data from reaching the O-DU for processing. This mismatch shows why cross-layer analysis is needed to correctly identify the root cause of the problem.

V. Results and Evaluation

This section compares FALCON++ performance against four baseline models: (i) **BGMM**, a statistical density-based approach; (ii) **Joint-VAEGAN (JVGAN)**, a generative deep learning model; (iii) **Simple AE**, a standard dense autoencoder; and (iv) **CONTEXT-AE**, a standard LSTM-Autoencoder with attention.

TABLE I: Global Performance Comparison: Statistical vs. Deep Learning Methods

Method Type	Model	Precision	Recall	F1-Score
Statistical	BGMM	0.57	**0.92**	0.71
Deep Learning	Simple AE	0.32	0.11	0.16
	Joint VAE (JVGAN)	0.52	0.07	0.13
	Context-AE (Ablation)	0.11	**0.92**	0.20
	FALCON++ (Ours)	**0.66**	**0.92**	**0.77**

The experimental results in Table I demonstrate that **FALCON++** significantly outperforms both statistical and deep learning baselines, achieving the highest F1-score of 0.77.

A critical observation is the "Precision Gap" between the ablation baseline (*Context-AE*) and our proposed framework. While *Context-AE* achieves a high Recall of 0.92, its low Precision (0.11) indicates an overwhelming number of False Positives, which would trigger unnecessary and costly mitigation actions in a production O-RAN environment.

In contrast, FALCON++ leverages feature-wise gating and residual skip connections to filter out ambient telemetry noise, improving Precision by 500% over the ablation model while maintaining maximum sensitivity. Furthermore, generative approaches like *JVGAN* exhibit a high False Negative rate (Recall of only 0.07), confirming that traditional reconstruction-based models often over-smooth sparse O-RAN anomalies.

By decomposing the network into logical CU-DU pairs, FALCON++ ensures topology scalability while enabling fault localization. Table II provides a granular breakdown of the detection recall across specific topological pairs. These results confirms that FALCON++ detects network-wide anomalies, though some CPU- and Memory-contentions are missed. However, it excels at anomaly localization by accurately finding the affected CU-DU pair. This is a critical requirement heading towards automated network healing in the long run as it helps in Root Cause Analysis (RCA).

Fig. 5 and Table III show that Expert Context AE, as employed by FALCON++, is architected for deployment scalability. Its lightweight design ensures that real-time anomaly detection is achieved without incurring significant system overhead, making it highly suitable for live O-RAN environments.

TABLE II: Fault Localization Performance of FALCON++ across CU-DU Pairs and Stress Types

CU-DU Pair / Fault Type	Recall	F1-score	Precision	Support
srscu0_srsdu0	1.00	0.82	0.69	308
srscu0_srsdu1	0.77	0.70	0.64	262
srscu1_srsdu2	1.00	0.78	0.64	238
CPU Stress	0.90	0.76	0.66	310
Memory Stress	0.87	0.74	0.65	244
Network Packet Loss	1.00	0.81	0.68	254

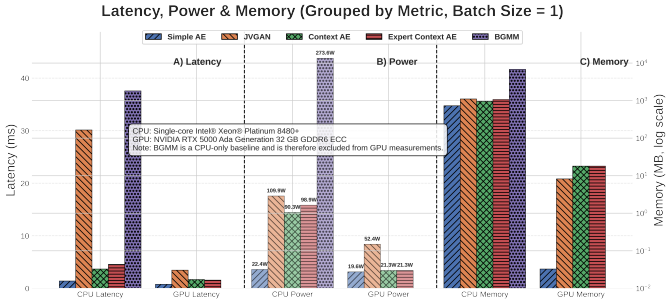

Fig. 5: Comparison of CPU/GPU latency, power, and memory for different anomaly detection models (batch size = 1).

Batch Size	CPU Time (ms)	GPU Time (ms)
2	2.4771	0.7393
8	3.2215	0.7325
16	4.9679	0.7255
64	15.1295	0.7345
256	55.3400	0.7492
1024	280.9896	2.1988
2048	699.6097	4.5410
4096	1281.4659 (> 1 sec)	9.3963
8192	—	19.1468
16384	—	42.2664
32768	—	84.8614

TABLE III: CPU vs GPU execution time for different batch sizes in EXPERT CONTEXT AE, as employed by FALCON++.

Moreover, the model scales efficiently with increasing batch sizes: while GPU deployment maintains low latency across a wide range of batches, CPU deployment remains efficient up to approximately 4000 pairs, beyond which an additional CPU instance is required to sustain performance. This demonstrates that EXPERT CONTEXT AE can handle increasing network scale while optimizing resource usage.

VI. CONCLUSIONS AND FUTURE WORK

The proposed **FALCON++** framework achieves high diagnostic fidelity in cloud-native O-RAN deployments, detecting subtle resource anomalies that traditional generative models often miss. Its pairwise CU-DU inferencing decomposes the network into logical links, ensuring linear scalability and precise fault localization. The FALCON++ attains an unsupervised F1-score of 0.77, while supervised benchmarks like XGBoost achieve a potential upper bound of 0.96 in topology-aware scenarios.

A. Future Work

Some potential research directions are as follows:

- **Closing the Accuracy Gap:** Boosting semi-supervised F1-score toward the 0.96 supervised benchmark.
- **Non-RT RIC–Based Fault Detection:** Implementing the proposed framework in the Non-RT RIC of O-RAN, enabling efficient identification of network faults while supporting AI/ML model training and long-term network optimization.
- **Explainable RCA (X-RCA):** Pinpointing software-level triggers or misconfigurations via feature-wise reconstruction errors.
- **RU-Level Monitoring Extension:** The current framework focuses only on CU–DU interactions and does not explicitly incorporate Radio Unit (RU) level telemetry. Future work could involve extending the monitoring pipeline to include RU-level KPIs, enabling finer-grained fault detection and localization across the full CU–DU–RU disaggregated O-RAN stack.

ACKNOWLEDGMENTS

This work was partially supported by the DST-EPSRC project *Synergy: Taking openness to the next level in 6G networks*, Grant number: DST/INT/UK/Telecom/P-181/2024(G) and the SERB, New Delhi, India, Grant number: JBR/2021/000005.

REFERENCES

[1] "Ai-ran alliance," AI-RAN Alliance, Tech. Rep., 2024. [Online]. Available: https://ai-ran.org/

[2] S. D. A. Shah, M. Hafeez, A. Salama, and S. A. R. Zaidi, "Proactive ai-and-ran workload orchestration in o-ran architectures for 6g networks," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 7939–7954, 2025.

[3] M. Mohammad, "Survey on ai-based reliability and anomaly detection in microservices," *International Journal of Computer Applications*, vol. 975, p. 8887, 2017.

[4] D. Raca, D. Leahy, C. J. Sreenan, and J. J. Quinlan, "Beyond throughput: The next generation: A 5g dataset with channel and context metrics," in *Proceedings of ACM MMSys*, 2020, pp. 1–12.

[5] D. Bega, M. Gramaglia *et al.*, "A dataset for 5g network traffic analysis," *Journal of Network and Computer Applications*, vol. 145, p. 102409, 2019.

[6] M. Sakurada and T. Yairi, "Anomaly detection using autoencoders with nonlinear dimensionality reduction," *Applied Intelligence*, vol. 48, no. 4, pp. 891–903, 2018.

[7] J. An and S. Cho, "Variational autoencoder based anomaly detection using reconstruction probability," *Special lecture on IE*, vol. 2, no. 1, pp. 1–18, 2015.

[8] P. V. A. Alves, M. A. S. S. Goldbarg *et al.*, "Machine learning-driven anomaly detection for 5g o-ran performance metrics," *Procedia Computer Science*, vol. 224, pp. 465–472, 2023.

[9] C. Sun, U. Pawar, M. Khoja, X. Foukas, M. K. Marina, and B. Radunovic, "Spotlight: Accurate, explainable and efficient anomaly detection for open ran," in *Proceedings of ACM MobiCom*, 2024.

[10] F. Benedetto *et al.*, "Robust procedural learning for anomaly detection and observability in 5g ran," *Journal of Network and Systems Management*, vol. 30, no. 3, pp. 1–28, 2022.

[11] Y. K. L S, S. Jain, B. R. Tamma, and K. Kondepu, "Falcon: A framework for fault prediction in open ran using multi-level telemetry," in *Proceedings of IEEE INFOCOM WKSHPS 2025*, pp. 1–6.

[12] "nodeexporter," https://github.com/prometheus/node_exporter, 2025, release 1.8.2.

[13] "cadvisor," https://github.com/google/cadvisor, 2025, release v0.49.2.