rApp Observability: Causal Attribution of KPI Changes to rApp Actions in O-RAN Non-RT RIC

rApp Observability: Causal Attribution of KPI Changes to rApp Actions in O-RAN Non-RT RIC

Roberto Riggio
Università Politecnica delle Marche, Ancona, Italy; Email: r.riggio@univpm.it

***Abstract*—Concurrent rApps (third-party applications running atop the Non-RT RIC) introduce a fundamental observability gap: when a Key Performance Indicator (KPI) changes, operators cannot systematically determine which rApp action caused it. Existing O-RAN conflict mitigation operates proactively or reactively within the control loop but offers no post-hoc causal attribution. We propose an observability architecture for the Non-RT RIC that captures timestamped rApp actions across the R1, A1, and O1 interfaces, aligns them with KPI time series, and applies a causal attribution methodology combining changepoint detection, Granger causality, conditional intervention effect, and temporal precedence scoring. Evaluation with three concurrent rApps over a simulated multi-cell 5G setup, across three scenarios of increasing difficulty, shows that the combined method achieves high accuracy when actions are temporally or spatially separated and consistently places the causal rApp within the top-2 candidates under concurrent overlapping actions on shared cells.**

***Index Terms*—O-RAN, Non-RT RIC, rApps, observability, causal attribution, Granger causality, changepoint detection**

I. Introduction

The O-RAN Non-Real-Time RAN Intelligent Controller (Non-RT RIC) hosts third-party applications, rApps, that optimize RAN behavior on timescales of seconds to minutes. While the Near-Real-Time RAN Intelligent Controller (Near-RT RIC) hosts xApps which operates on timescales between ten milliseconds and one second [1]. With commercial platforms already integrating tens of rApps from multiple vendors and the O-RAN Software Community (OSC) providing an open-source reference implementation, multi-rApp deployments are becoming the norm rather than the exception. This proliferation creates a pressing observability problem. In the Near-RT RIC, the Conflict Mitigation Framework (CMF) can intercept all control messages at a single point before they reach the RAN [2]. No equivalent mechanism exists at the Non-RT RIC level. rApp actions are dispersed across heterogeneous interfaces: A1 policies are forwarded to the Near-RT RIC and take effect indirectly through xApp-mediated E2 control messages, while O1 configuration changes modify RAN node parameters directly without any xApp involvement. The R1 interface that connects rApps to the Non-RT RIC framework is service-oriented and does not provide a centralized interception point where all outgoing actions could be inspected and correlated. Furthermore, the longer timescale of the Non-RT RIC control loop (> 1 s, compared to 10 ms–1 s for the Near-RT RIC) widens the temporal gap between an rApp action and its observable KPI effect, making causal attribution inherently more ambiguous. When a KPI shifts and multiple rApps have acted within overlapping time windows on a shared cell scope, the operator currently has no systematic means to determine which action caused the change.

Existing work includes game-theoretic resolution [3], distillation-based methods for direct inter-xApp conflicts [4], digital twins-based approaches [5] and graph neural networks [6]–[8]. PACIFISTA showed that uncoordinated applications can cause substantial instability and significant performance degradation [9]. QACM explicitly addresses post-action conflict detection and mitigation in live networks, leveraging KPI monitoring to detect and resolve conflicts at runtime [10]. xApp-level studies analyze how conflicts manifest in KPIs during operation to enable post-hoc conflict detection and classification [11]. AI-driven frameworks incorporate reactive detection and classification pipelines [12]. These works, however, do not address the finer-grained problem of attributing an observed KPI anomaly to a ranked set of rApp action instances in the Non-RT RIC. This is the gap filled by this paper.

This paper makes three contributions. First, a non-intrusive observability architecture for the Non-RT RIC that captures timestamped rApp actions across the R1, A1, and O1 interfaces and aligns them with KPI time series. Second, a causal attribution methodology combining changepoint detection [13], Granger causality [14], conditional intervention effect (CIE) estimation, and temporal precedence scoring into a composite ranking of candidate causes. Third, an evaluation with three concurrent rApps on a simulated multi-cell 5G setup, showing high accuracy under separated actions and consistent top-2 identification of the causal rApp under concurrent overlapping actions on shared cells.

II. Observability Architecture and Causal Attribution Methodology

A. Components

The observability layer is a dedicated rApp on the Non-RT RIC consuming R1 services in read-only mode, with four components. The *Action Logger* subscribes to R1 service invocations and records structured events capturing timestamp, originating rApp identifier, action type (A1 policy create/update/delete, O1 Configuration Management (CM) parameter modification, or Data Management and Exposure (DME) write), target scope (cell, slice, or node), and parameter name with old and new values. The *KPI Collector* subscribes to O1 performance measurements and stores per-cell counters at a configurable granularity, tagging each sample with the Action Logger's time base for alignment. The *Correlation Engine*

implements the attribution pipeline: it consumes the action log and KPI series from a shared time-series store, performs changepoint detection on each stream, matches changepoints against candidate rApp actions, and produces ranked results. The *Attribution Reporter* exposes these results via the R1 DME service, associating each KPI anomaly with a ranked list of candidate rApp actions and confidence scores.

B. Data Model and Preprocessing

1) KPI Feature Set: The KPI Collector produces, for each cell c and reporting period Δ, a feature vector of five counters: cell downlink throughput (Mbps); Physical Resource Block (PRB) utilization ratio; handover success rate; handover failure count; and average serving-cell RSRP (dBm). These are selected because each is directly affected by the rApps under study: load balancing acts primarily on PRB utilization and throughput, energy saving on throughput and PRB utilization via carrier deactivation, and mobility robustness optimization on handover success rate and failure count. The methodology is not restricted to this set: the pipeline treats each counter as an independent time series, so additional counters can be included at a computational cost that scales linearly with the number of counters. In operational deployments, the KPI set should cover the union of counters affected by the deployed rApps, since omitting a mediating KPI may increase the temporal lag between action and detected anomaly, reducing attribution confidence.

2) Preprocessing: KPI counter values are z-score normalized per cell and per counter using the running mean and standard deviation computed over a trailing window of 300 s, chosen to be approximately $5\times$ the lag window $\delta = 60$ s, long enough that individual rApp actions do not contaminate the running statistics yet short enough to track gradual environmental drift. This serves two purposes: it ensures that the PELT changepoint detector and the CIE estimator operate on comparable scales across cells with different baseline traffic levels, and it acts as a local detrending operation that absorbs slow-moving trends (e.g., gradual load increases) into the rolling statistics. The VAR models used for Granger causality testing are consequently fitted on the normalized, locally detrended series rather than on raw KPI values, reducing the risk of spurious correlations arising from shared non-stationary trends between action indicators and KPI counters.

3) Action Windows: The Correlation Engine operates on temporally aligned pairs of action sequences and KPI segments. We define an *action window* $w_i = [t_i, t_{i+1})$ as the interval between consecutive actions from any rApp on a given target scope, where t_i is the timestamp of the i-th action and t_{i+1} that of the next action on the same scope. The corresponding *KPI observation window* is the KPI time-series segment $\mathbf{K}(t)$ for $t \in w_i$ restricted to the KPI counters associated with that scope. This pairing ensures that each KPI segment is influenced by at most one rApp action at its leading edge, simplifying the attribution logic in the sequential case. When multiple actions fall within the same observation window, the Correlation Engine applies the multi-candidate scoring procedure described in Section II-F.

C. Problem Formulation

Let $\mathcal{R} = \{r_1, \ldots, r_N\}$ be the set of N concurrently active rApps, and let $\mathcal{A} = \{a_1, a_2, \ldots\}$ be the temporally ordered sequence of actions, where each action $a_j = (t_j, r_j, s_j, p_j)$ records a timestamp t_j, the originating rApp $r_j \in \mathcal{R}$, a target scope s_j (set of affected cells), and the modified parameter–value pair p_j. Let $\mathbf{K}_c(t)$ denote the multivariate KPI time series for cell c sampled at discrete intervals Δ. Given a detected KPI anomaly at time t^* on cell c^*, the goal is to produce a ranked list of candidate actions $\{(a_j, \phi_j)\}$ where $\phi_j \in [0, 1]$ is a confidence score reflecting the likelihood that action a_j caused the anomaly. We define an anomaly as a statistically significant shift in the distribution of a KPI k on cell c, detected at time τ by a changepoint detection algorithm.

D. Stage 1: Intervention Detection

The first stage identifies statistically significant KPI shifts that may reflect rApp actions. We apply the Pruned Exact Linear Time (PELT) algorithm independently to each KPI counter of each cell, using a penalized likelihood cost with the modified Bayesian Information Criterion (mBIC). PELT detects a set of changepoints $\mathcal{C}_c^k = \{\tau_1, \tau_2, \ldots\}$ for each cell c and KPI counter k. Not every changepoint corresponds to an rApp action; exogenous events (e.g., propagation changes) also produce KPI shifts. To filter these, we retain only changepoints that fall within a configurable lag window δ of at least one logged action whose target scope includes cell c:

$$\mathcal{C}_c^{k,\text{filt}} = \left\{\tau \in \mathcal{C}_c^k \;\middle|\; \exists\, a_j \in \mathcal{A} : c \in s_j \;\wedge\; 0 < \tau - t_j \leq \delta\right\}. \quad (1)$$

E. Stage 2: Candidate Action Matching

Each retained changepoint defines a candidate anomaly event (τ, c, k), for which we retrieve the set of actions:

$$\mathcal{A}(\tau, c) = \left\{a_j \in \mathcal{A} \;\middle|\; c \in s_j \;\wedge\; 0 < \tau - t_j \leq \delta\right\}. \quad (2)$$

If $|\mathcal{A}(\tau, c)| = 1$, attribution is unambiguous and the single candidate receives $\phi = 1$. When $|\mathcal{A}(\tau, c)| > 1$, the causal scoring stage is invoked to rank the candidates.

F. Stage 3: Causal Scoring

1) Granger Causality Score: For each rApp r_i, we construct a binary action indicator series $\mathbf{x}_i(t)$, where $\mathbf{x}_i(t) = 1$ if rApp r_i acted on cell c at time t, and 0 otherwise. We test whether $\mathbf{x}_i$ Granger-causes $\mathbf{K}_c^k$ by fitting two vector autoregressive (VAR) models of order L: a *restricted* model that predicts $\mathbf{K}_c^k(t)$ from its own past values, and an *unrestricted* model that additionally includes lagged values of $\mathbf{x}_i$ [14]. The model of order L is selected using the Bayesian Information Criterion (BIC), i.e., a fixed L is used across all cells. The Granger score is defined as:

$$g_i = 1 - p_i, \quad (3)$$

where p_i is the p-value of the F-test comparing the restricted and unrestricted model residuals. To mitigate false positives from indirect causal paths, we use conditional Granger causality, conditioning on the action indicator series of all other rApps when testing each r_i.

2) Conditional Intervention Effect: The Granger score captures statistical precedence over the entire observation period but does not isolate the effect of a specific action instance. To estimate the local causal effect of a candidate action a_j on the anomaly at τ, we employ a difference-in-differences (DiDs) estimator. For a given cell c and KPI counter k, we define the pre-action mean $\bar{K}_c^{\text{pre}}$ as the average of $K_c^k(t)$ over the $W = \lfloor w/\Delta \rfloor$ samples in the interval $(t_j - w,\, t_j]$, and the post-action mean $\bar{K}_c^{\text{post}}$ as the average over the W samples in $(t_j,\, t_j + w]$, where t_j is the action timestamp, Δ the KPI reporting interval, and w the window duration. Let c' be a control cell, i.e., a cell not in the target scope s_j but sharing a similar baseline KPI profile with c^*, selected as the cell outside s_j that minimizes the Euclidean distance between its pre-action KPI mean vector and that of c^*. In practice, candidate control cells are restricted to first-order neighbors of c^* (i.e., cells whose coverage areas are adjacent) to ensure comparable propagation and traffic conditions. The CIE is:

$$\text{CIE}_j = \big(\bar{K}_{c^*}^{\text{post}} - \bar{K}_{c^*}^{\text{pre}}\big) - \big(\bar{K}_{c'}^{\text{post}} - \bar{K}_{c'}^{\text{pre}}\big). \tag{4}$$

We normalize by the standard deviation of the pre-action KPI on c^* to obtain a dimensionless score:

$$\hat{e}_j = \min\Big(1,\ \frac{|\text{CIE}_j|}{\sigma_{c^*}^{\text{pre}}}\Big). \tag{5}$$

The cap at 1 ensures that $\hat{e}_j$ is bounded to the same $[0, 1]$ range as g_i and $\hat{d}_j$, preventing large-magnitude interventions from dominating the composite score and rendering the Granger and temporal components irrelevant.

3) Temporal Precedence Score: The two preceding methods capture causal evidence through statistical testing and local effect estimation, respectively, but neither accounts for the temporal proximity of an action to the observed anomaly. Since rApp-induced KPI changes typically manifest within seconds to tens of seconds of the action, proximity carries complementary, though not independently sufficient, discriminative value. We encode it as a decay-based score:

$$\hat{d}_j = \exp\big(-\lambda\,(\tau - t_j)\big), \tag{6}$$

where $\lambda > 0$ is a decay rate parameter. Actions immediately preceding the changepoint receive scores close to 1, while distant actions decay toward 0. This score carries no causal information on its own, an action that immediately precedes an exogenous KPI shift would receive a high $\hat{d}_j$ despite being unrelated, and is therefore always used in combination with the Granger and CIE components, never in isolation.

4) Score Aggregation: Each scoring method addresses a limitation of the others, motivating their combination. Granger causality captures long-run statistical precedence but operates over the entire observation period, without pinpointing which specific action caused a given anomaly. The CIE estimator fills this gap by isolating the local effect of an individual action, but requires a suitable control cell and is sensitive to concurrent environmental trends that differencing may not fully remove. The temporal precedence score adds a complementary proximity signal that discriminates between candidates when Granger and CIE yield comparable scores at different temporal distances from the changepoint; on its own, however, it carries no causal information and performs poorly in isolation (as confirmed in Section III). The composite formulation retains the strengths of all three at modest additional cost.The three component scores are combined via a weighted sum:

$$\phi_j = \alpha\, g_{r_j} + \beta\, \hat{e}_j + \gamma\, \hat{d}_j, \tag{7}$$

where $\alpha + \beta + \gamma = 1$. The scores operate at two levels of granularity: the Granger score g_i is computed per rApp r_i, while the CIE score $\hat{e}_j$ and the temporal score $\hat{d}_j$ are computed per individual action a_j. The notation g_{r_j} resolves this by mapping each candidate action a_j to the Granger score of its originating rApp. This two-level design is intentional: the Granger score acts as a prior on whether the rApp is causally relevant at all, while the CIE and temporal scores disambiguate which specific action instance from that rApp is responsible for a particular anomaly. The weights can be set uniformly ($\alpha = \beta = \gamma = 1/3$) or tuned on a validation set if labeled attribution data is available. Candidates are ranked by decreasing ϕ_j.

G. Handling Concurrent Actions

When two or more actions from different rApps target the same cell within an interval shorter than δ, individual attribution reaches a fundamental identifiability limit: the KPI observation window contains the superimposed effects of multiple interventions. Rather than forcing a single attribution, the method flags the event as *ambiguous* when the gap between the top-ranked and second-ranked candidate satisfies $\phi_1 - \phi_2 < \epsilon$, where ϵ is a configurable ambiguity threshold. Ambiguous events are reported with a shared attribution across the top candidates, accompanied by their individual scores. This design reflects the practical reality that truly simultaneous actions on identical scope cannot be disentangled without additional information such as parameter-level causal models.

III. Experimental Evaluation

A. Testbed and rApp Deployment

We deploy the observability rApp on the OSC Non-RT RIC alongside three concurrent rApps representing potentially conflicting optimization objectives: *LB-rApp* (load balancing) adjusts Cell Individual Offset (CIO) parameters via A1 policies to redistribute handovers and balance PRB utilization; *ES-rApp* (energy saving) deactivates capacity carriers during low-traffic periods via O1 CM changes and reactivates them as load increases; and *MRO-rApp* (mobility robustness optimization) tunes handover hysteresis and time-to-trigger (TTT) via A1 policies to reduce radio link failures and ping-pong handovers. The radio environment is simulated with ns-3 and the 5G-LENA module. Table I summarizes the key parameters.

The ns-3 simulation and the Non-RT RIC interact through a custom layer that bridges the simulated RAN with the O-RAN interfaces. On the O1 path, the adapter extracts per-cell KPI counters from ns-3 internal statistics every $\Delta = 1\,\text{s}$ and publishes them as VES (Virtual Event Streaming) notifications at 1-second granularity. On the A1 path, the adapter translates A1 policies received from the Non-RT RIC into corresponding ns-3 parameter modifications: CIO values are mapped to the handover offset attributes of the A3 event

TABLE I
SIMULATION AND METHODOLOGY PARAMETERS.

Parameter	Value
Macro sites / cells	7 / 21
Inter-site distance	500 m
Carrier freq. (coverage / capacity)	3.5 GHz / 3.6 GHz
Bandwidth per carrier / PRBs	100 MHz / 273
Numerology (μ) / SCS	1 / 30 kHz
Tx power / antenna gain	43 dBm / 15 dBi
Antenna panel	8×8 dual-pol, 65° HPBW, 6° tilt
gNB height / UE height	25 m / 1.5 m
Propagation / fading	3GPP UMa (TR 38.901) / CDL-C
Indoor UE ratio / penetration loss	20% / 20 dB
Noise figure (gNB / UE)	5 dB / 7 dB
UE count / mobility	200 / random waypoint, 3–30 km/h
Traffic model	CBR DL UDP, 5 Mbps per UE
MAC scheduler	Proportional fair
RLC mode / HARQ retx	UM / 3
Baseline HO hysteresis / TTT	3 dB / 160 ms
Warm-up / simulation duration	120 s / 3600 s
Independent runs per scenario	10
KPI granularity (Δ)	1 s
Lag window (δ)	60 s
VAR order (L)	10 (BIC-selected median over $[1, 20]$)
CIE window (w)	30 s
Temporal decay (λ)	0.05 s^{-1}
Composite weights (α, β, γ)	1/3 each
Ambiguity threshold (ϵ)	0.1

model, and hysteresis/TTT updates are applied to the relevant `LteUeRrc` parameters. On the O1 CM path, carrier activation and deactivation commands are translated into enabling or disabling secondary component carriers on the target cells.

B. Ground Truth and Metrics

Because the simulation is fully controlled, ground-truth attribution is possible: ns-3 logs trace each rApp action and its direct KPI impact. Each run includes 60 rApp actions (20 per rApp). For every detected anomaly, the ground truth identifies the responsible rApp or labels it exogenous if no rApp caused the shift. We report five metrics. *Attribution accuracy* (Acc) is the fraction of anomalies for which the top-ranked candidate matches the ground truth. *Top-k accuracy* (Acc@k) is the fraction for which the ground truth appears among the top-k candidates. *False positive rate* (FPR) is the fraction of attributions pointing to a non-causal rApp. *Detection latency* is the wall-clock time from anomaly detection to attribution output. *Computational overhead* is the CPU and memory consumption of the observability rApp during the run.

C. Scenario S1: Sequential Actions

In S1, rApps act one at a time with a minimum inter-action interval of 90 s, ensuring that each KPI observation window is influenced by at most one action. This baseline validates the end-to-end pipeline under unambiguous conditions.

Results confirm near-perfect attribution. Across 10 independent runs, the combined scoring method achieves Acc $= 0.97 \pm 0.02$, with the residual errors caused by a small number of exogenous KPI shifts that coincide with the lag window of an rApp action. Each individual method also performs well in this setting: Granger alone reaches 0.95, CIE alone 0.96, and temporal precedence alone 0.93. The PELT changepoint detector identifies 0.93 of the true KPI shifts injected by rApp actions, with a false discovery rate of 0.05. Average detection latency is 1.2 s per anomaly event.

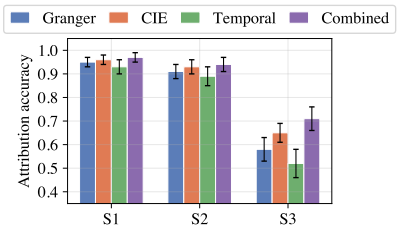

Fig. 1. Attribution accuracy of individual methods.

D. Scenario S2: Overlapping Actions, Disjoint Scope

In S2, rApps act concurrently, i.e., multiple actions may occur within the same lag window, but each action targets a distinct cell. This tests whether per-cell scoping resolves ambiguity when actions overlap temporally but not spatially.

Attribution accuracy remains high: Acc $= 0.94 \pm 0.03$. Per-cell candidate matching (Eq. 2) effectively isolates most events, since each cell's KPI segment is influenced by only one rApp. The small accuracy drop relative to S1 is primarily due to inter-cell interference effects: when ES-rApp deactivates a carrier on cell c_1, displaced traffic shifts to neighboring cell c_2, producing a secondary KPI change on c_2 that the changepoint detector attributes to a different rApp acting on c_2 within the lag window. The CIE estimator partially mitigates this because its DiDs formulation cancels spillover components that are shared between the target and control cells, but the cancellation is imperfect when the spillover magnitude varies across neighbors. A practical refinement for operational deployments would be to extend each action's target scope s_j to include first-order neighbors, preventing spillover-induced changepoints from being matched to unrelated rApp actions at the cost of slightly larger candidate sets. Fig. 1 (left group) shows the per-method and combined accuracy for S1 and S2.

E. Scenario S3: Overlapping Actions, Shared Scope

S3 is the hardest case: rApps act concurrently on the same cells, with inter-action intervals as short as 5 s on shared scope. This tests the limits of causal attribution when multiple interventions are superimposed on the same KPI segment.

Top-1 accuracy degrades to Acc $= 0.71 \pm 0.05$, reflecting the fundamental identifiability limit discussed in Section II-G. However, the combined scoring method consistently outperforms each individual method: Granger alone drops to 0.58, CIE to 0.65, and temporal precedence to 0.52. The composite score benefits from complementary information, producing attributions that are more robust than any single method.

Top-k accuracy reveals that the correct rApp almost always appears among the leading candidates: Acc@2 $= 0.91 \pm 0.03$ and Acc@3 $= 0.97 \pm 0.02$. Fig. 2 shows the top-k curves for S3. The ambiguity detector flags 24% of S3 events as ambiguous ($\phi_1 - \phi_2 < \epsilon$); among these, 89% have the true cause within the shared-attribution set, confirming that the

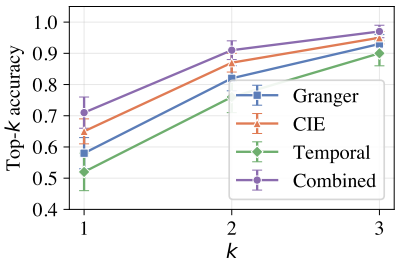

Fig. 2. Top-k attribution accuracy for Scenario S3.

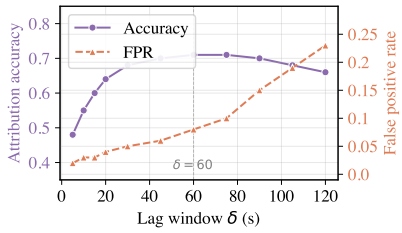

Fig. 3. Attribution accuracy and false positive rate in Scenario S3.

ambiguity mechanism captures genuine identifiability limits rather than masking errors.

F. Sensitivity and Overhead Analysis

1) Lag Window Sensitivity: We vary δ from 5 s to 120 s on S3 and measure attribution accuracy and false positive rate. Fig. 3 shows the trade-off. At $\delta = 5$ s, the candidate set misses many true causal actions because their KPI effects have not yet materialized, yielding low attribution accuracy (Acc $= 0.48$). As δ increases, accuracy improves and plateaus around $\delta =$ 45–75 s. Beyond $\delta = 90$ s, FPR increases noticeably (> 0.15) as unrelated actions enter the candidate set. The default $\delta =$ 60 s sits at the favorable operating point.

2) Weight Sensitivity: We sweep the composite weight vector (α, β, γ) over a grid with step 0.1, subject to $\alpha+\beta+\gamma = 1$, and evaluate Acc on S3. The best-performing configuration is $(\alpha, \beta, \gamma) = (0.3, 0.4, 0.3)$, which slightly favors the CIE component and achieves Acc $= 0.74$, a modest improvement over uniform weighting (0.71). Accuracy remains above 0.65 for all configurations where $\beta \geq 0.2$, indicating that the CIE estimator is the most informative single component under concurrent overlapping actions.

3) Computational Overhead: The observability rApp runs as a single-threaded Python process with no parallelization; all three scoring stages execute sequentially per anomaly event. CPU and memory usage are sampled at 1 s intervals. The observability rApp consumes 120 MB of memory and an average of 0.3 vCPU. To identify the computational bottleneck, we instrument the Correlation Engine with per-stage wall-clock timers. Across all anomaly events in S3, VAR model fitting for the Granger causality stage accounts for approximately 70% of total per-event computation time, candidate matching accounts for 5%, CIE estimation for 20%, and temporal precedence scoring plus aggregation for the remaining 5%. Detection latency averages 1.2 s per event across all scenarios, well within the Non-RT RIC's operational timescale (> 1 s).

IV. Conclusions

This paper addressed the problem of post-hoc causal attribution of KPI changes to rApp actions in the O-RAN Non-RT RIC. Experimental evaluation with three concurrent rApps over a simulated 21-cell 5G scenario demonstrated that the combined scoring method achieves attribution accuracy above 0.94 when rApp actions are temporally or spatially separated, and that even under concurrent overlapping actions on shared cells, i.e., the hardest case, the causal rApp appears within the top-2 ranked candidates in over 90% of events.

Several directions remain for future work. First, extending the attribution chain across RIC tiers would enable end-to-end cross-layer observability. Second, reconstructing causal chains across rApps, where one rApp's action triggers a KPI change that drives another rApp to act, would extend the method from proximate to root cause attribution, potentially leveraging R1 service registration metadata to constrain the search over candidate chains. Finally, introducing a minimum confidence threshold below which the method rejects attribution and classifies an anomaly as exogenous would reduce false positives when KPI shifts coincide with rApp actions by chance rather than causation.

References

[1] M. Polese, L. Bonati, S. D'Oro, S. Basagni, and T. Melodia, "Understanding o-ran: Architecture, interfaces, algorithms, security, and research challenges," *IEEE Communications Surveys & Tutorials*, vol. 25, no. 2, pp. 1376–1411, 2023.

[2] C. Adamczyk and A. Kliks, "Conflict mitigation framework and conflict detection in o-ran near-rt ric," *IEEE Communications Magazine*, vol. 61, no. 12, pp. 199–205, 2023.

[3] A. Wadud, F. Golpayegani, and N. Afraz, "Conflict management in the near-rt-ric of open ran: A game theoretic approach," in *Proc. of IEEE iThings*, 2023, pp. 479–486.

[4] H. Erdol, X. Wang, R. Piechocki, G. Oikonomou, and A. Parekh, "xapp distillation: Ai-based conflict mitigation in b5g o-ran," *Computer Networks*, vol. 274, p. 111848, 2026.

[5] A. E. Giannopoulos, S. T. Spantideas, G. Levis, A. S. Kalafatelis, and P. Trakadas, "Comix: Generalized conflict management in o-ran xapps—architecture, workflow, and a power control case," *IEEE Access*, vol. 13, pp. 116 684–116 700, 2025.

[6] A. Zolghadr, J. F. Santos, L. A. DaSilva, and J. Kibilda, "Learning and reconstructing conflicts in o-ran: A graph neural network approach," in *Proc. of IEEE WCNC*, Milan, Italy, 2025.

[7] M. A. Shami, J. Yan, and E. T. Fapi, "O-ran xapps conflict management using graph convolutional networks," 2025. [Online]. Available: https://arxiv.org/abs/2503.03523

[8] A. Zolghadr, J. F. Santos, L. A. DaSilva, and J. Kibiłda, "Learning and reconstructing conflicts in o-ran: A graph neural network approach," 2025. [Online]. Available: https://arxiv.org/abs/2412.14119

[9] P. Brach del Prever, S. D'Oro, L. Bonati, M. Polese, M. Tsampazi, H. Lehmann, and T. Melodia, "Pacifista: Conflict evaluation and management in open ran," *IEEE Transactions on Mobile Computing*, vol. 24, no. 10, pp. 10 590–10 605, 2025.

[10] A. Wadud, F. Golpayegani, and N. Afraz, "Qacm: Qos-aware xapp conflict mitigation in open ran," *IEEE Transactions on Green Communications and Networking*, vol. 8, no. 3, pp. 978–993, 2024.

[11] ——, "xapp-level conflict mitigation in o-ran, a mobility driven energy saving case," in *Prof. of IEEE INFOCOM*, London, UK, 2025.

[12] A. Wadud, N. Afraz, and F. Golpayegani, "Ai-powered conflict management in open ran: Detection, classification, and mitigation," 2026. [Online]. Available: https://arxiv.org/abs/2602.19758

[13] R. Killick, P. Fearnhead, and I. A. Eckley, "Optimal detection of changepoints with a linear computational cost," *Journal of the American Statistical Association*, vol. 107, no. 500, pp. 1590–1598, 2012.

[14] C. W. J. Granger, "Investigating causal relations by econometric models and cross-spectral methods," *Econometrica*, vol. 37, no. 3, pp. 424–438, 1969.