ClosRCA-Bench: An Open Topology-Grounded Benchmark and Counterfactual Recovery Framework for Self-Healing Datacenter Networks

Dheeraj Ramasahayam
Independent Researcher
United States

***Abstract*—Open research on datacenter-network root cause analysis (RCA) is constrained by private production traces and by public studies that omit the topology or remediation context needed for self-healing systems. This paper introduces *ClosRCA-Bench*, a reproducible topology-grounded benchmark built from Cisco's public Clos-topology telemetry by combining event files, CDP maps, and per-device YANG telemetry into fixed graph windows. The benchmark contains 311 windows over 11 topology nodes with 30 features per node and four cause families: BFD outage, blackhole, ECMP change, and interface shutdown. Two fault classes localize to hidden target devices that are not directly monitored, making topology-aware localization central. We compare rule-based, correlation-based, graph-free, and spatio-temporal graph RCA methods, and we evaluate remediation through a safety gate and a counterfactual topology digital twin. On the held-out split, Random Forest achieves anomaly F1 0.9688 and cause F1 0.9707, while the full STGNN reaches 0.8380 weighted target-device F1 and 1.0000 hidden-target accuracy; removing topology reduces hidden-target accuracy to 0.0000. These results establish an open benchmark for topology-aware, temporal, and recovery-aware RCA.**

***Index Terms*—datacenter networks, root cause analysis, anomaly detection, graph neural networks, network telemetry, self-healing systems, benchmark datasets**

I. Introduction

Datacenter networks underpin cloud computing, artificial intelligence training, digital public services, and financial infrastructure. When failures occur, operators need more than alarms; they need fast attribution and a response path that is both effective and safe. This is difficult in practice because symptoms propagate across the topology, measurements are often incomplete, and many remediation actions can make the incident worse if applied to the wrong device.

Recent work has shown that large-scale cloud and datacenter fault localization can be automated to a meaningful degree [1], [2]. Graph-based RCA methods have also become more effective at reasoning over dependencies and alert propagation [3], [4]. The remaining open problem is not merely model design. It is how to study topology-aware RCA and self-healing behavior on public, reproducible data rather than only on synthetic traces or private incident corpora.

This paper should also be read as deliberately distinct from earlier predictive-maintenance work, including the author's prior attention-guided failure forecasting study [5]. That earlier work asked whether telemetry can predict failure within a forward warning horizon. The present work addresses a different operational question: once abnormal behavior is present, can a topology-grounded system localize the likely target and recommend an action that survives safety and counterfactual recovery checks?

The paper makes four concrete contributions:

- *ClosRCA-Bench*, a topology-grounded public benchmark constructed from Cisco Clos-topology telemetry scenarios, event files, and CDP maps [6], [7],
- a spatio-temporal graph RCA model with explicit topology and temporal ablations for hidden-target localization,
- temporal root-cause tracking, compound-failure slicing, and a public case-study workflow that expose delay and propagation rather than only static labels,
- a safety-gated and counterfactual remediation protocol that measures action correctness, service recovery, and unsafe-action blocking on held-out anomalous windows.

II. Related Work

Production-oriented datacenter fault detection and localization have been studied in both academic and industrial settings. Roy *et al.* presented passive real-time fault localization for datacenters [1], while Li *et al.* described automated incident detection for cloud services [2]. These efforts established the operational importance of fast localization, but the underlying data is difficult to reproduce publicly.

Graph-based RCA has become a natural direction because infrastructure failures propagate over dependencies rather than isolated time series. AlertRCA showed that causality-enhanced graph learning improves alert-based RCA [3], and graph convolutional networks provide an effective message-passing foundation for this class of problems [4]. At the same time, public network telemetry repositories from Cisco have enabled anomaly-detection studies on event-linked network data [6], [7]. In operational practice, however, RCA systems still range from hand-authored rules to correlation engines to graph models. The evaluation in this paper therefore compares a rule-based heuristic, a correlation/template baseline, a graph-only baseline, and the full temporal graph model rather than treating topology-aware learning in isolation.

The present work is positioned as a benchmark-and-systems paper rather than a claim of a wholly new backbone architecture. Its novelty lies in combining public topology metadata,

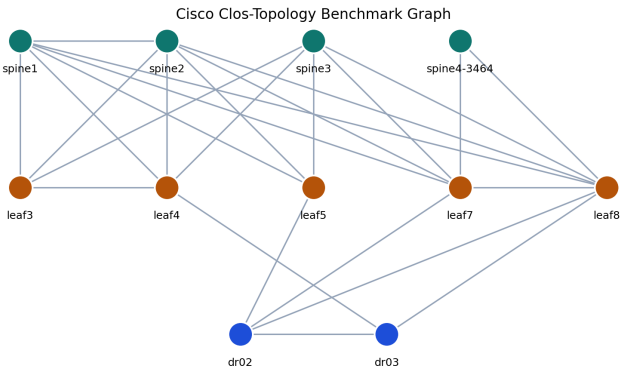


Fig. 1. Topology graph used by the public Cisco Clos-topology benchmark. Hidden-target labels include `leaf3` and `spine4-3464`.

TABLE I
BENCHMARK SUMMARY

Property	Value
Scenarios	8
Cause families	4
Graph nodes	11
Features per node	30
Window size	6 bins (60 s)
Window step	3 bins (30 s)
Total windows	311
Normal windows	147
Anomalous windows	164
Hidden target devices	2

hidden-target RCA, and counterfactual remediation validation into one reproducible workflow.

III. Benchmark Construction

ClosRCA-Bench is built from eight public scenarios in Cisco's Clos-topology telemetry collection [7]. Each selected scenario includes an event file, a CDP map, and per-device YANG telemetry archives for the monitored subset of devices. We retain four fault families: `bfd_outage`, `blackhole`, `ecmp_change`, and `interface_shutdown`.

The construction pipeline performs four steps. First, it parses per-device telemetry archives and extracts compact device-level features from data-rate, generic-counter, BFD, BGP, FIB-drop, interface-state, and CPU collections. Second, it aligns samples into 10-second bins and builds a fixed graph whose nodes are routers in the Clos fabric. Third, it labels anomaly intervals directly from the event files. Fourth, it converts each scenario into 60-second graph windows with a 30-second step.

Two target devices, `leaf3` and `spine4-3464`, are not directly present in the monitored telemetry subset. They are nevertheless retained as localization labels because the topology exposes their adjacency to the observed nodes. This makes hidden-target localization measurable rather than implicit.

The benchmark supports three tasks:

- binary anomaly detection over graph windows,
- multiclass cause classification over anomalous windows,
- target-device localization over anomalous windows.

In addition, the released evaluator converts window-level outputs into scenario-level temporal traces. For each test scenario, it aligns the first detected anomalous window with the fault-onset timestamp in the event file to report detection delay in seconds. It also projects post-fault node activations onto the topology graph to recover a causal chain from the inferred root target to downstream affected nodes.

IV. Model and Recovery Twin

The proposed RCA model is a spatio-temporal graph neural network (STGNN). Each graph window is a tensor of shape (T, N, F), where T is time, N is the number of topology nodes, and F is the per-node feature count. A GRU encodes each node's temporal behavior, graph convolution propagates state across the topology, and three heads predict anomaly, cause class, and target device.

The target-localization head is intentionally topology-aware. Instead of predicting only from a pooled graph embedding, it scores the candidate target nodes directly after message passing. This is important for hidden-target scenarios because the candidate device may have no direct telemetry but can still be inferred from neighboring symptoms.

Two ablations are evaluated:

- **No topology**: replace the graph adjacency with the identity matrix,
- **No temporal**: replace temporal encoding with a mean-pooled projection.

The paper also evaluates two strong graph-free baselines, Random Forest and MLP, on flattened graph windows, plus a rule-based RCA heuristic and a correlation/template baseline. The graph-only GCN baseline is implemented as the no-temporal ablation, which removes the GRU while keeping graph propagation. This gives a direct comparison between static graph reasoning and the proposed temporal graph model.

Remediation is evaluated in two stages. First, a safety gate validates whether an action is allowed on the predicted target and whether it respects simple topology constraints such as alternate-path availability for interface reroutes. Second, a lightweight topology digital twin scores the counterfactual effect of the action on service reachability, path diversity, overload, and blast radius. Each predicted cause-target pair maps to an action template such as `restore_bfd_session`, `rollback_blackhole_route`, or `reroute_and_restore_interface`. This converts remediation from advisory text into a measurable recovery stage.

V. Results

Table II reports the held-out benchmark metrics. Random Forest is strongest overall on anomaly detection, cause classification, and aggregate target-device localization. This is an important benchmark result in its own right: the public benchmark is not a guaranteed deep-learning win. The STGNN nevertheless shows a clear topology-specific advantage on the

hidden-target slice, which is the benchmark setting where topology should matter most.

The topology effect is clearest in target localization. On the full anomalous test slice, the STGNN improves target-device weighted F1 from 0.6566 in the no-topology ablation to 0.8380. On the hidden-target subset, the full model reaches 1.0000 accuracy while the no-topology ablation falls to 0.0000. This indicates that message passing is doing useful work specifically where direct telemetry is unavailable.

Table III makes the temporal-tracking contribution explicit. The rule-based and correlation baselines can react quickly on simple signatures, but their RCA accuracy remains weak. The graph-only baseline matches the full STGNN's RCA accuracy but lags in detection delay, improving from 6.3 s to 5.2 s only when temporal state is retained. Because all four methods remain below 0.05 ms per window once features are materialized, the dominant operational tradeoff here is not raw inference speed but delay-accuracy behavior under topology-aware reasoning.

Compound-failure scenarios are the more realistic stress test. On single-failure windows the STGNN reaches 1.0000 cause accuracy; on compound-failure windows it falls to 0.9130. The graph-only baseline matches this compound-failure accuracy but again detects later, which suggests that temporal state helps most in converting topology-aware RCA from a static labeler into a fast incident tracker rather than merely a higher-scoring classifier. Because the public mixed-event slice is still modest, we also add a supplementary 59-node synthetic Clos stress study with 900 windows, masked hidden targets, and simultaneous faults such as routing disturbance plus interface shutdown. In that stress study, the STGNN retains 0.8788 simultaneous-fault cause accuracy, compared with 0.8485 for Random Forest.

A. Why a Graph Model?

Table IV addresses the obvious question raised by Table II: if Random Forest wins aggregate cause F1 on the public benchmark, why use the more complex graph model at all? The answer is that the graph model is not the best choice for every aggregate metric on a small fixed topology. It is the more defensible RCA engine when topology-dependent reasoning is the actual requirement. In the supplementary 59-node stress study, the full STGNN reaches 1.0000 hidden-target accuracy while Random Forest reaches 0.8846, and the STGNN degrades less under simultaneous faults. It also produces propagation traces directly, whereas the tabular baseline remains a labeler without a causal chain.

The synthetic scale-up study is not presented as a replacement for the public benchmark. Its purpose is narrower: to test whether the same RCA design remains interpretable and operationally lightweight when the topology grows from 11 nodes to a larger 59-node Clos fabric and the fault process becomes more entangled. Figure 2 shows that even at this larger scale, batch inference stays below 0.12 ms per window and above 8.4k windows/s for the full STGNN.

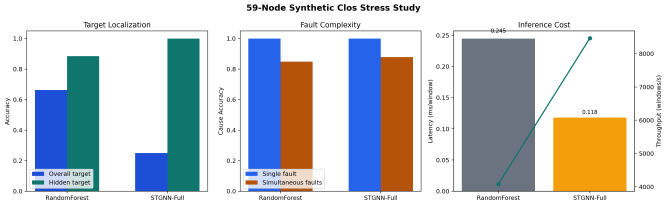


Fig. 2. Supplementary 59-node synthetic Clos stress study showing hidden-target localization, simultaneous-fault accuracy, and inference cost.

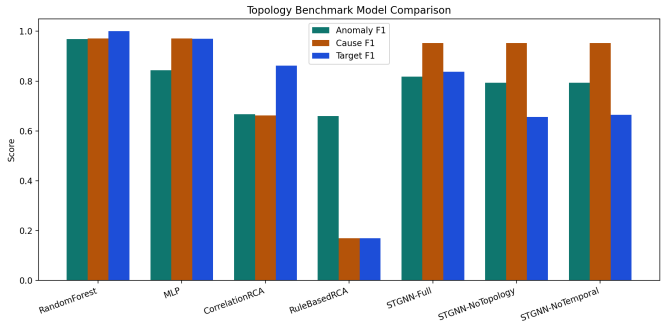


Fig. 3. Held-out comparison of anomaly, cause, and target metrics on the topology-grounded benchmark.

The remediation layer reaches 0.8485 action-match rate, 0.8485 safety-pass rate, and blocks 100% of mismatched unsafe actions on held-out anomalous windows. In the counterfactual recovery twin, the same gated actions improve mean service reachability from 0.9740 in the injected-fault state to 1.0000 after recovery, reduce mean blast radius by 0.0260, and achieve 0.8182 recovery-success rate. In other words, the gate does not merely mirror the classifier. It suppresses bad actions, and the digital twin translates that suppression into measurable recovery outcomes.

B. Real Failure Case Study

We also include a concrete public case study based on scenario `S-200202_2014_evtmix-1`. This trace was collected from a controlled internal lab environment simulating enterprise traffic. The injected fault is an interface shutdown on `leaf4`. The full STGNN identifies both the correct cause and the correct target, and it does so 23.3 s after the logged fault onset. The recovered propagation chain shows downstream activation on `leaf8` after 23.3 s, `leaf7` after 83.3 s, and `spine3` after 143.3 s. This turns the benchmark from a static classification exercise into a systems-oriented RCA trace with a recoverable causal chain.

C. Deployment and IDS Integration

Figure 7 shows the intended deployment path. The RCA engine sits downstream from telemetry ingestion and optional anomaly or IDS signals, and upstream from operator display or remediation validation. This is the systems connection to the author's prior intrusion-detection work [5]: an IDS or anomaly detector can surface the incident window, after which

TABLE II
HELD-OUT BENCHMARK PERFORMANCE

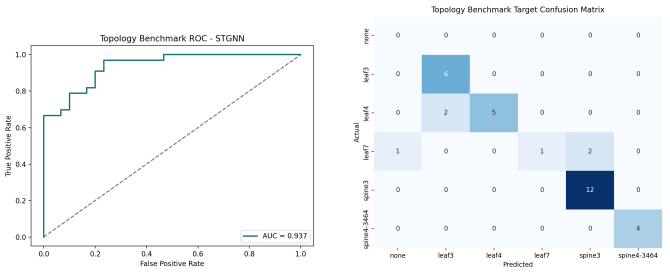
Model	Anomaly F1	Cause $F1_w$	Target $F1_w$	Hidden-Target Accuracy	Unsafe Blocked
Random Forest	0.9688	0.9707	1.0000	1.0000	–
MLP	0.8438	0.9707	0.9702	1.0000	–
STGNN Full	0.8182	0.9532	0.8380	1.0000	1.0000
STGNN No Topology	0.7931	0.9524	0.6566	0.0000	–
STGNN No Temporal	0.7931	0.9532	0.6646	0.9000	–

TABLE III
TEMPORAL TRACKING AND RCA POSITIONING

Method	RCA Acc.	Delay (s)	Latency (ms)
Rule-based RCA	0.3030	39.1	0.0009
Correlation RCA	0.6364	1.6	0.0074
GCN-Static	0.9394	6.3	0.0064
STGNN Full	0.9394	5.2	0.0422

TABLE IV
WHY USE A GRAPH MODEL?

Capability	RF	STGNN
Public cause $F1_w$	0.9707	0.9532
59-node hidden-target acc.	0.8846	1.0000
59-node simult. cause acc.	0.8485	0.8788
Propagation traces	No	Yes



(a) STGNN ROC (b) Target confusion matrix

Fig. 4. Held-out anomaly and target-localization diagnostics for the full topology model.

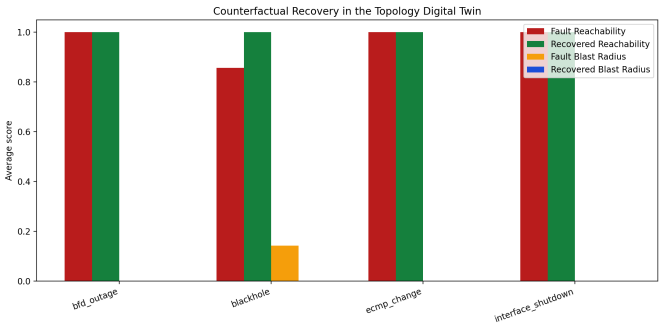


Fig. 5. Counterfactual recovery results in the topology digital twin, grouped by fault family.

the temporal RCA engine localizes the root target, traces propagation, and forwards a cause-target pair to the safety gate and digital twin. In this sense, *ClosRCA-Bench* is not only a benchmark but also a deployable middle layer between detection and response.

VI. DISCUSSION AND LIMITATIONS

Three observations matter most. First, benchmark construction is the primary scientific contribution here. The value is not that the STGNN beats every baseline; it is that topology-aware and recovery-aware RCA can now be studied on public data with fixed artifacts and hidden-target labels. Second, the topology signal appears where it should: not in easy aggregate classification, but in localizing devices without direct telemetry. Third, the recovery twin makes remediation measurable. A paper about self-healing should report not only whether a classifier is correct, but whether the proposed action improves network state under a counterfactual replay.

The current benchmark still has clear limits. It covers only eight scenarios from one lab topology. The network-loop scenario in the public repository was not included in the fixed benchmark split because the unpacked directory does not expose the per-device telemetry archives needed by the current parser. The compound-failure slice is therefore realistic but still modest; it reflects mixed-event operational traces rather than fully simultaneous multi-root incidents. The new 59-node synthetic scale-up study partially addresses scale and simultaneous-fault complexity, but it is still synthetic and therefore not a substitute for more public production-like traces. The recovery twin is a lightweight topology simulator rather than a controller-in-the-loop emulator such as containerlab or Batfish. Finally, strong classical baselines remain difficult to beat on the aggregate metrics, so future work should focus on harder scenarios, richer labels, live-control validation, and explicitly overlapping multi-failure traces.

VII. CONCLUSION

This work provides, to our knowledge, the first public, topology-grounded, recovery-aware RCA benchmark for datacenter networks. It reframes the project around a stronger and more defensible thesis: public topology-grounded benchmarking plus temporal RCA tracking and counterfactual recovery validation for self-healing datacenter networks. *ClosRCA-Bench* exposes hidden-target localization, supports reproducible anomaly and RCA evaluation, reports detection delay and propagation traces, and measures remediation quality explicitly through both safety gating and a topology digital

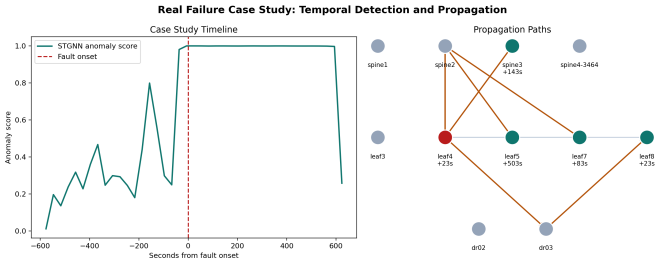


Fig. 6. Public lab case study showing temporal detection and graph-based propagation tracing for the `leaf4` interface shutdown scenario.

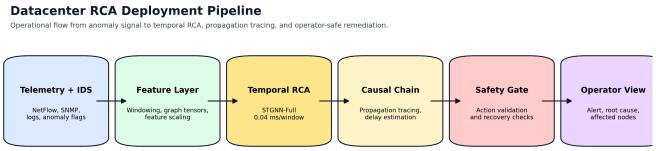


Fig. 7. Deployment view for temporal RCA in a self-healing datacenter workflow.

twin. The topology-aware STGNN is most useful where topology should matter most, namely hidden-target localization, simultaneous-fault tracking, and fast temporal tracking, while the recovery layer blocks unsafe actions and quantifies service restoration. The next step is to expand the benchmark with additional public scenarios, more severe multi-root failures, and live integration with emulated or production-like control loops.

REFERENCES

[1] A. Roy, H. Zeng, J. Bagga, and A. C. Snoeren, "Passive realtime datacenter fault detection and localization," in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*. USENIX Association, 2017, pp. 595–612. [Online]. Available: https://www.usenix.org/conference/nsdi17/technical-sessions/presentation/roy

[2] L. Li, X. Zhang, X. Zhao, H. Zhang, Y. Kang, P. Zhao, B. Qiao, S. He, P. Lee, J. Sun, F. Gao, L. Yang, Q. Lin, S. Rajmohan, Z. Xu, and D. Zhang, "Fighting the fog of war: Automated incident detection for cloud systems," in *2021 USENIX Annual Technical Conference (USENIX ATC 21)*. USENIX Association, 2021, pp. 131–146. [Online]. Available: https://www.usenix.org/conference/atc21/presentation/li-liqun

[3] Z. Yu, Q. Ouyang, C. Pei, X. Wang, W. Chen, L. Su, H. Jiang, X. Wang, J. Li, and D. Pei, "Causality enhanced graph representation learning for alert-based root cause analysis," in *2024 IEEE 24th International Symposium on Cluster, Cloud and Internet Computing (CCGrid)*. IEEE, 2024, pp. 77–86.

[4] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *International Conference on Learning Representations (ICLR)*, 2017. [Online]. Available: https://openreview.net/forum?id=SJU4ayYgl

[5] D. Ramasahayam, "Temporal attention-guided sequence learning for zero-shot generalization in datacenter network failures," 2026, unpublished manuscript on early-warning failure prediction.

[6] A. Putina, D. Rossi, A. Bifet, S. Barth, D. Pletcher, C. Precup, and P. Nivaggioli, "Telemetry-based stream-learning of BGP anomalies," in *Proceedings of the ACM SIGCOMM 2018 Workshop on Big Data Analytics and Machine Learning for Data Communication Networks*, 2018, pp. 15–21.

[7] Cisco Innovation Edge, "Telemetry: Open-source datasets for network anomaly machine learning and research," https://github.com/cisco-ie/telemetry, 2026, gitHub repository accessed March 16, 2026.