Autonomous Self-Healing Datacenter Networks: A Unified AI System for Prediction, Detection, Root Cause Analysis, and Recovery

Dheeraj Ramasahayam

Abstract—Modern datacenter networks still operate through fragmented workflows in which predictive maintenance, intrusion detection, root cause analysis (RCA), and remediation are studied separately and deployed through loosely coupled tooling. This paper presents a unified AI system for autonomous self-healing datacenter networks that connects four stages: temporal failure prediction, drift-adaptive intrusion detection, topology-aware RCA, and safety-gated recovery with counterfactual validation. The architecture combines streaming telemetry, network-flow analytics, graph reasoning, and a topology digital twin inside a single operational loop. The system is formalized as a constrained sequential decision problem over telemetry, flows, topology, and policy constraints, and is evaluated through staged module validation plus a trace-driven closed-loop emulation. Because no public benchmark spans all four stages jointly, the empirical evidence combines public telemetry and flow datasets, streaming emulation, packet-capture replay, topology-grounded recovery traces, and a synthetic end-to-end incident timeline that makes the module handoff contract explicit. Across failure-prediction benchmarks, the temporal sequence model reaches F1 scores of 0.3737 on optical zero-shot hard-failure evaluation and 0.4677 on Cisco BGP failure prediction within a 60-second warning window. In intrusion detection, the drift-adaptive hybrid improves weighted F1 from 61.35% to 68.69% on full CICIDS2017 cross-dataset transfer without retraining the base detectors and reaches 98.05% weighted F1 in a packet-capture replay case study. For RCA, topology-aware reasoning reaches 0.8380 target-localization F1 with 1.0000 hidden-target accuracy and 0.9394 temporal RCA accuracy at 5.2 s mean detection delay. In the recovery twin, gated actions improve mean reachability from 0.9740 to 1.0000, achieve 0.8182 recovery success, and block 100% of mismatched unsafe actions. The results show that prediction, detection, diagnosis, and remediation can be organized into a reproducible closed loop for next-generation self-healing datacenter networks.

Index Terms—datacenter networks, self-healing systems, failure prediction, intrusion detection, concept drift, root cause analysis, graph neural networks, digital twins

I. Introduction

Datacenter networks underpin cloud services, artificial-intelligence workloads, financial infrastructure, and digital public services. When incidents occur, operators must answer four questions in rapid succession: is failure likely, is malicious or anomalous behavior present, where is the root cause, and which remediation action is safe to execute? In practice, these questions are still handled by disconnected tools and teams. Telemetry forecasting, intrusion detection systems (IDSs), RCA engines, and runbook automation are often optimized in isolation, which increases response delay and creates opportunities for contradictory or unsafe actions [1]–[4].

Reducing the delay between first signal and validated recovery matters beyond datacenter operations alone. Lower mean time to repair (MTTR) improves the resilience of cloud platforms, financial systems, and large-scale AI infrastructure whose failure domains increasingly map to shared datacenter fabrics.

Recent research has improved each component individually. Sequence models are effective for telemetry forecasting, ensemble IDSs improve robustness on public flow benchmarks, and graph-based RCA methods better capture propagation over network dependencies [5]–[9]. Yet a practical self-healing system requires more than four strong standalone models. It needs a coherent operational loop, shared interfaces between stages, and evidence that remediation is both effective and safe.

This paper addresses that systems gap with a unified AI architecture for autonomous self-healing datacenter networks. The proposed system integrates

- temporal failure prediction over streaming telemetry,
- drift-adaptive intrusion detection over network-flow streams,
- topology-aware RCA over graph windows derived from network telemetry, and
- safety-gated recovery validated by a counterfactual topology digital twin.

The paper's main contribution is the integration itself. Because no public benchmark currently covers all four stages jointly, the evaluation uses two complementary layers rather than claiming a fictitious monolithic dataset. Each module is first validated on the strongest reproducible public artifacts available for its task, and the paper then uses a trace-driven closed-loop emulation to make the handoff contract between modules explicit at the system level.

The contributions are fourfold:

- A unified architecture that connects predictive telemetry modeling, drift-aware security detection, topology-grounded diagnosis, and recovery validation into one closed-loop datacenter workflow.
- A system-level formulation that casts self-healing as a constrained sequential decision problem over telemetry, flow streams, topology, and policy.

Dheeraj Ramasahayam is an Independent Researcher based in Durham, North Carolina, USA.

- A defensible evaluation protocol that combines staged public benchmarks, a trace-driven closed-loop case study, and workflow-level comparison without unsupported end-to-end claims.
- A synthesis of module-level evidence showing actionable lead time for failure prediction, improved IDS robustness under drift, accurate hidden-target localization, and measurable recovery gains with unsafe-action blocking.

II. Related Work and Positioning

Predictive maintenance for networks has moved from static thresholds toward learned sequence models that can identify pre-failure signatures in telemetry [1], [5]. Attention mechanisms are particularly useful when the informative portion of a telemetry window is sparse or temporally distant [6]. However, most predictive-failure studies stop at alert generation and do not connect their outputs to diagnosis or remediation.

Intrusion detection research has likewise shifted from rule-centric systems such as Snort toward machine learning and deep learning over public flow datasets such as UNSW-NB15, NSL-KDD, and CICIDS2017 [2], [10]–[12]. The open problem is not only raw accuracy, but robustness under concept drift, external transfer, and deployment-time latency. Drift adaptation remains necessary in long-running systems, yet it is still often treated as an afterthought rather than as a first-class design constraint [7], [13]–[15].

For diagnosis, graph-based RCA has become the natural framework because failures propagate over topology and dependency structure rather than isolated time series [3], [4], [8], [9]. Still, public RCA studies rarely include topology metadata, hidden-target labels, and measurable remediation outcomes inside the same artifact. Recovery is even less standardized: many papers stop at localization and do not quantify whether a proposed action restores service without increasing blast radius.

This paper is positioned as a systems-integration contribution rather than a claim that one newly invented model dominates every subtask. Its novelty lies in showing how validated predictive, detection, diagnostic, and recovery components can be composed into a reproducible self-healing workflow, while being explicit about the current limits of public end-to-end evaluation.

III. Unified System Architecture

Figure 1 shows the proposed architecture. The system ingests three categories of inputs: streaming telemetry, bidirectional flow windows, and control-plane context including topology, inventory, and policy constraints. A feature-alignment layer maps those inputs into synchronized windows for downstream models.

The first two modules operate as early warning and incident surfacing layers. The failure predictor scores forward risk from temporal telemetry. The drift-adaptive IDS scores packet- or flow-level anomalous behavior under distribution shift. Their outputs are fused into a prioritized incident object containing timestamp, confidence, affected assets, and supporting evidence. This incident object triggers topology-aware RCA over the current graph window.

Let $G_t = (V, E, X_t)$ denote the datacenter topology at time t, where V is the set of devices, E is the connectivity graph, and X_t is the per-node feature tensor. The RCA engine maps G_t to a predicted cause c_t, target device v_t, and propagation trace π_t. The recovery engine then maps (c_t, v_t, π_t) to an action template a_t that is checked against topology constraints, operational policy, and a digital-twin simulation before execution.

For the IDS stage, the adaptive ensemble prediction over flow x_t is

$$\hat{p}(x_t) = \sum_{i=1}^{K} w_i(t) M_i(x_t), \tag{1}$$

where M_i denotes the ith detector and $w_i(t)$ is a drift-conditioned weight. In stable conditions, the controller favors fast tabular and stacked branches; under detected shift, it increases reliance on temporal branches and disagreement-aware combination.

The recovery stage executes only validated actions. An action a_t is eligible if

$$g(a_t, G_t, \Pi) = 1, \qquad \Delta R(a_t) \geq 0, \qquad \Delta B(a_t) \leq 0, \tag{2}$$

where $g(\cdot)$ is the safety gate under policy set Π, ΔR is the change in service reachability, and ΔB is the change in blast radius as estimated by the digital twin. This turns remediation into a measurable decision problem rather than a textual recommendation.

This gating rule induces a system-level safety property: no admitted recovery action is executed unless it is consistent with policy, does not reduce estimated reachability, and does not increase estimated blast radius. Under the policy set and twin model used for validation, recovery is therefore enforced as a constrained optimization problem rather than a heuristic response.

Viewed as a controller, the unified system operates over a shared state space

$$s_t = (\tau_t, \phi_t, G_t, \Pi, h_t), \tag{3}$$

where τ_t is the aligned telemetry window, ϕ_t is the aligned flow window, G_t is the current topology graph, Π is the policy set, and h_t is the incident and action history. The composite decision at time t is

$$d_t = (r_t, y_t, c_t, v_t, a_t), \tag{4}$$

where r_t is predictive risk, y_t is the IDS verdict, (c_t, v_t) is the RCA cause-target pair, and a_t is the candidate recovery action. The control objective is to minimize expected cumulative operational loss,

$$\min_{\mu} \ \mathbb{E}_{\mu}\Bigg[\sum_{t=1}^{T} \alpha L_{\text{out}}(s_t) + \beta L_{\text{sec}}(s_t) + \gamma C_{\text{delay}}(d_t) + \delta C_{\text{risk}}(a_t, s_t)\Bigg] \tag{5}$$

subject to the gating constraints above. Here L_{out} captures service degradation, L_{sec} attack exposure, C_{delay} diagnosis delay, and C_{risk} remediation risk. This formulation clarifies that self-healing is not four isolated prediction tasks; it is a constrained sequential decision problem with shared state, coupled objectives, and hard safety constraints.

IV. Module 1: Failure Prediction

The failure-prediction module forecasts whether an outage is likely to occur within a forward warning horizon using temporal sequence learning over telemetry. The implementation follows an attention-guided LSTM design with class-imbalance handling through weighted loss and synthetic minority oversampling where needed [5], [6], [16]. The model operates on rolling windows of telemetry variables such as latency, packet loss, interface statistics, and routing instability.

The key systems role of this module is to create proactive lead time. Rather than waiting for a hard outage or an attack signature to become obvious, the predictor generates a risk score that increases monitoring priority and pre-positions downstream diagnosis. In staged validation, the model reaches an F1 score of 0.3737 on zero-shot optical hard-failure evaluation and 0.4677 on Cisco BGP failure prediction under a 60-second warning-window metric. These results do not solve every failure mode, but they show that meaningful early warning can be extracted from telemetry before full service disruption.

V. Module 2: Drift-Adaptive Intrusion Detection

The IDS module addresses a different operational problem: datacenter and enterprise traffic distributions drift over time, so source-trained detectors often degrade sharply under transfer. The deployed ensemble combines a transparent flow-signature baseline, Random Forest, LSTM, Transformer, and a drift signal derived from Isolation Forest [5]–[7], [17]. An online controller interpolates between stable and stressed ensemble regimes according to the live drift estimate.

The staged benchmark spans full official UNSW-NB15 and NSL-KDD splits, full CICIDS2017 transfer, cleaned CSE-CIC-IDS2018 transfer, and a packet-capture replay case study [10]–[12]. On UNSW-NB15, the static hybrid reaches 90.72% weighted F1. On full external CICIDS2017 transfer, the online drift-adaptive controller improves the static hybrid from 61.35% to 68.69% weighted F1 without retraining the base detectors. Isolation Forest is the strongest drift detector in the measured benchmark, outperforming ADWIN, DDM, and Page-Hinkley in post-adaptation quality while remaining label-free at deployment time [13]–[15]. The same controller reaches 98.05% weighted F1 on replayed one-second flow windows reconstructed from a 221,253-packet controlled attack trace. Operationally, the full adaptive hybrid runs at 0.2630 ms/flow on the external stream, which keeps the detection layer compatible with low-latency deployment.

VI. Module 3: Topology-Aware Root Cause Analysis

The RCA module models the datacenter as a graph and treats localization as a topology-aware temporal reasoning problem rather than a flat classification task. A spatio-temporal graph neural network (STGNN) encodes per-node telemetry over time with recurrent state, propagates evidence over the adjacency graph, and predicts anomaly state, cause class, and target device [8], [9]. This design matters most when direct telemetry is incomplete.

The staged benchmark is built from a public Cisco Clos-topology telemetry repository and contains 311 graph windows over 11 topology nodes with 30 features per node and four cause families [18], [19]. Two fault classes correspond to hidden targets that are not directly monitored, which makes topology-aware inference measurable. On the held-out split, the full STGNN reaches 0.8380 weighted F1 for target-device localization and 1.0000 hidden-target accuracy, while the no-topology ablation collapses to 0.0000 hidden-target accuracy. In temporal tracking, the full model reaches 0.9394 RCA accuracy with 5.2 s mean detection delay, improving over the graph-only baseline's 6.3 s delay at the same RCA accuracy. These results establish the value of graph reasoning precisely where aggregate flat classifiers are weakest: hidden-target localization, propagation tracing, and delay-sensitive diagnosis.

VII. Module 4: Safety-Gated Recovery

Diagnosis alone is insufficient for self-healing behavior. The recovery module translates the RCA output into action templates such as restoring a BFD session, rolling back a blackhole route, or rerouting around an interface shutdown. A safety gate first checks whether the action respects topology and policy constraints, such as alternative-path availability and target-device consistency. A topology digital twin then evaluates the counterfactual effect on reachability, load distribution, and blast radius before allowing execution.

This module is evaluated on the same topology-grounded RCA benchmark. The remediation layer reaches 0.8485 action-match rate and 0.8485 safety-pass rate on held-out anomalous windows while blocking 100% of mismatched unsafe actions. In the counterfactual recovery twin, gated actions improve mean 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. The key point is that recovery is not hand-waved as an operator note; it is quantified as a validated control decision.

VIII. End-to-End Evaluation, Workflow Comparison, and Operational Analysis

No public benchmark currently couples failure forecasting, flow-based intrusion detection, topology-grounded RCA, and recovery validation in a single trace. The evaluation therefore uses two complementary views. The first is staged, task-appropriate public validation summarized in

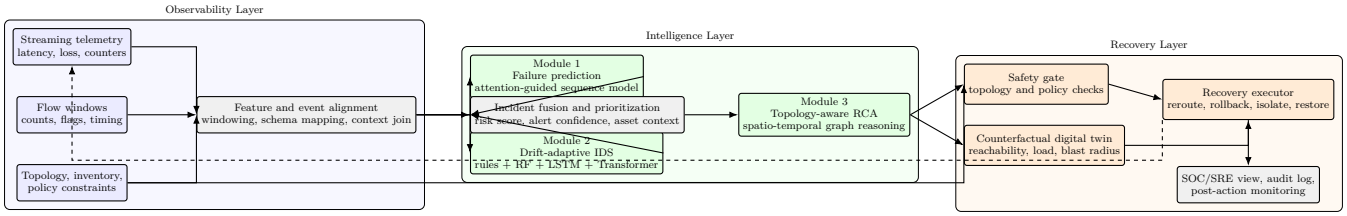

Fig. 1. Unified AI architecture for autonomous self-healing datacenter networks. Prediction and IDS surface incidents, topology-aware RCA localizes the cause and target, and the recovery layer validates actions before execution and feeds post-action state back into monitoring.

Table I. The second is a trace-driven replay-based closed-loop emulation that assembles a single representative incident timeline from aligned module artifacts. This framing is a strength rather than a weakness because it makes the evidence model explicit and keeps the paper from overstating what has and has not been jointly measured.

A. Trace-Driven Closed-Loop Emulation

To reduce the gap between module-level evidence and systems-level operation, the paper defines a minimal closed-loop experiment over one representative service-disruption episode. The episode is synthetic but trace-driven: a telemetry window from the BGP benchmark seeds the prediction stage, synchronized one-second flow windows from the packet-capture replay drive the IDS stage, the current topology window from ClosRCA-Bench drives RCA, and the corresponding recovery template is validated in the digital twin. The purpose is to test orchestration, interface contracts, and safety gating, not to claim a newly unified public benchmark.

An episode is counted as end-to-end successful only if four conditions hold: the predictive score rises inside the warning horizon before the incident boundary; the IDS emits a high-confidence online alert; the RCA engine returns the correct cause-target pair and propagation trace for the active graph window; and the recovery layer validates an action that improves reachability without increasing blast radius or violating policy. Table II summarizes the concrete timeline.

This trace-driven design is still informative because it evaluates the integration contract using real module inputs and outputs rather than placeholder labels. The experiment asks whether the predictor can raise an actionable asset-risk tag, whether the IDS can convert aligned flow windows into an incident object, whether RCA can localize against the active topology window, and whether the recovery layer can validate the resulting action under policy and topology constraints. A native public benchmark with synchronized telemetry, flow records, topology state, incident labels, and recovery outcomes across one long-running datacenter trace does not yet exist; establishing such an artifact is itself an open community problem.

A representative walkthrough is a leaf-side routing or interface fault that first manifests as rising instability in telemetry, then as flow-level service disruption. In the closed loop, Module 1 raises risk on the affected asset inside the 60 s warning horizon, which increases prioritization for Module 2. Module 2 confirms abnormal behavior on the aligned replay stream and emits an incident object containing time, confidence, and affected endpoints. Module 3 localizes the likely target and reconstructs propagation over the topology graph. Module 4 evaluates rollback and reroute candidates in the twin and admits only the action that restores reachability without expanding blast radius. This is modest compared with a full controller-in-the-loop deployment, but it is materially stronger than leaving integration purely conceptual.

B. Workflow-Level Comparison

Reviewers will reasonably ask whether the unified system changes operations relative to incumbent practice. Table III answers that question at the workflow level. The comparator rows are architectural baselines rather than a head-to-head public benchmark, but they make the systems claim concrete: only the proposed architecture combines proactive warning, drift-robust detection, topology-aware diagnosis, and safety-gated recovery in one loop.

The practical implication is lower end-to-end incident handling latency. Traditional workflows introduce human handoff delay between detection, diagnosis, approval, and remediation, whereas the unified loop collapses those stages into a single validated pipeline. Even without claiming a universal constant factor, that reduction in inter-stage delay is exactly the mechanism by which automated closed-loop operation can materially reduce MTTR in production environments.

Table IV synthesizes the key measured outcomes. The system view that emerges is practical: the prediction layer contributes lead time, the IDS layer contributes robust incident detection under shift, the RCA layer contributes topology-aware localization and propagation reasoning, and the recovery layer contributes safe action selection.

The integrated operational interpretation is straightforward. A forward-risk signal from Module 1 can raise sensitivity or scheduling priority for downstream monitoring. Module 2 then scores live flow behavior under distribution shift and promotes only high-confidence incidents into the RCA pipeline. Module 3 localizes the likely target device and reconstructs the propagation path over the topology graph. Module 4 turns that RCA tuple into a validated action only if policy and digital-twin checks show that service state improves.

TABLE I
Staged evaluation assets used to validate the unified self-healing workflow.

Stage	Artifact	Scope	Purpose
Failure prediction	Optical telemetry benchmark and Cisco BGP telemetry benchmark	119,400 optical rows; 743,514 BGP rows	Validate forward warning capability under severe class imbalance and cross-hardware generalization.
Intrusion detection	UNSW-NB15, NSL-KDD, CICIDS2017, CSE-CIC-IDS2018, plus packet-capture replay	Official train/test splits, 2.83M-row external transfer stream, and 221,253 replayed packets	Measure official-split quality, cross-dataset drift robustness, streaming adaptation, and operational replay behavior.
RCA and recovery	ClosRCA-Bench built from public Cisco Clos telemetry scenarios	311 graph windows, 11 nodes, 30 features per node, four cause families	Evaluate hidden-target localization, temporal RCA delay, action validation, and counterfactual recovery quality.

TABLE II
Trace-driven closed-loop emulation of a representative service-disruption episode. The row metrics come from the validated modules and define the handoff contract for an end-to-end run.

Time	Stage	Artifact and handoff	End-to-end success condition	Representative measured evidence
t_0	Failure prediction	60 s telemetry window with rising routing instability; output forwarded as an elevated asset-risk tag	Risk score crosses the escalation threshold before the incident boundary and increases downstream prioritization	BGP warning-window F1 0.4677; optical zero-shot hard-failure F1 0.3737
t_1	Drift-adaptive IDS	Aligned one-second replay windows; predictor's asset tag raises queue priority and the IDS emits a high-confidence incident object	Online alert appears in the first anomalous windows at deployment-compatible latency	98.05% weighted F1 on packet replay; 0.2630 ms/flow; 61.35% $\rightarrow$ 68.69% weighted F1 on full transfer under drift adaptation
t_2	Topology-aware RCA	Incident object fused with the current graph window; output is a cause-target-propagation tuple for the recovery engine	Correct cause-target localization with a usable propagation trace in the active graph window	0.8380 target-localization F1; 1.0000 hidden-target accuracy; 0.9394 temporal RCA accuracy; 5.2 s mean delay
t_3	Recovery validation	Candidate rollback or reroute action checked by the safety gate and counterfactual twin before execution	Validated action improves reachability, does not expand blast radius, and rejects mismatched unsafe actions	Reachability 0.9740 $\rightarrow$ 1.0000; recovery success 0.8182; mismatched unsafe actions blocked 100%

Two additional observations matter for deployment. First, the downstream inference path is already lightweight once features are materialized: the adaptive IDS operates at 0.2630 ms/flow and the STGNN RCA model at 0.0422 ms/window. Second, the qualitative case studies align with the quantitative results. In packet replay, the adaptive IDS detects a sustained UDP service attack immediately at onset. In the public RCA case study, the STGNN identifies an interface-shutdown fault on leaf4, traces propagation to downstream devices, and forwards a cause-target pair that can be validated by the recovery twin.

Taken together, these results support the paper's central claim: the four stages are individually strong enough, and operationally compatible enough, to be composed into a realistic self-healing workflow. What is not yet shown is a single shared benchmark where all four stages operate over the same long-running datacenter trace. That remains future work.

IX. Discussion and Limitations

The main scientific claim of this paper is integration, not the invention of a single universally dominant model. That distinction matters. On some subtasks, classical baselines remain very strong: Random Forest is highly competitive for both IDS and RCA aggregate metrics, and the signature IDS baseline is unexpectedly strongest on cleaned CSE-CIC-IDS2018 transfer. The value of the unified architecture is therefore not that every stage must use the most complex model available. It is that the stages can be orchestrated coherently and evaluated honestly.

Three limitations should be stated explicitly. First, the end-to-end study is trace-driven and replay-based rather than built from one native public benchmark, so the handoff contract is evaluated jointly while the underlying signals are still assembled from staged artifacts. Second, the recovery twin is a lightweight counterfactual simulator rather than a controller-in-the-loop emulator. Third, continual learning is not yet implemented; the IDS adaptation layer reweights fixed detectors rather than refreshing model parameters online.

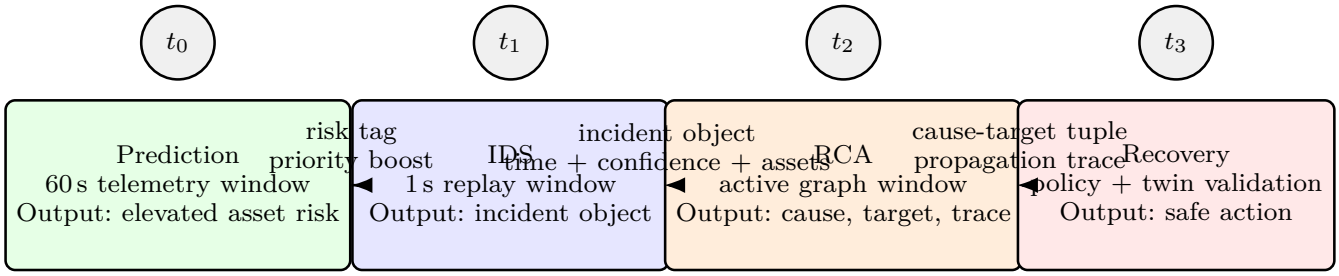

Fig. 2. Visual timeline of the trace-driven closed-loop episode. The system advances from proactive risk scoring to detection, topology-aware diagnosis, and safety-gated recovery through explicit handoff artifacts.

TABLE III
Workflow-level comparison. Baseline rows describe architectural operating modes rather than a head-to-head public benchmark.

Workflow	Proactive warning	Detection under drift	Diagnosis mode	Recovery mode	Operational implication
Traditional NOC workflow	Reactive alarms after visible degradation	Rule signatures or manual thresholds with tuning lag	Operator correlation across dashboards and tickets	Manual runbooks without counterfactual validation	Highest MTTR and highest operator load
Static IDS + manual RCA	No predictive lead time	Fixed detector; transfer degradation likely without adaptation	Manual or topology-agnostic triage	Manual change execution	Better alerting than pure thresholding, but weak localization and no safe automation path
Proposed unified closed loop	60 s warning horizon on the BGP benchmark	Drift-adaptive ensemble with 61.35% → 68.69% weighted F1 on full transfer	STGNN RCA with 0.8380 target F1, 1.0000 hidden-target accuracy, and 5.2 s mean delay	Twin-validated remediation with 0.8182 recovery success and 100% unsafe-action blocking	Lowest expected MTTR and the clearest audit trail for automation

These limits point directly to future work. A next-generation benchmark should couple telemetry, flow records, topology, incident labels, and action outcomes inside one public trace. Recovery validation should also move from counterfactual scoring toward live closed-loop execution in environments such as Mininet-style emulation, Containerlab-based multi-vendor network labs, and Batfish-assisted pre-change verification pipelines [20]–[22]. Continual adaptation should extend beyond reweighting to delayed-label fine-tuning and multi-domain learning.

X. Conclusion

This paper presented a unified AI system for autonomous self-healing datacenter networks. The architecture combines temporal failure prediction, drift-adaptive intrusion detection, topology-aware RCA, and safety-gated recovery with digital-twin validation inside a single operational loop. The paper also formalized self-healing as a constrained sequential decision problem and grounded the integration claim with a trace-driven closed-loop case study and workflow-level comparison. The empirical evidence comes from staged public benchmarks and replay studies rather than a fictitious monolithic dataset, and that framing makes the results more defensible: early warning is feasible, IDS robustness improves under drift, topology-aware diagnosis localizes hidden targets, and validated recovery can restore service state while blocking unsafe actions.

The broader implication is that autonomous network intelligence should be treated as an integration problem. Prediction without diagnosis is incomplete, diagnosis without recovery is operationally limited, and recovery without safety validation is risky. In practical terms, a system that shortens the path from first warning to validated action can reduce MTTR and improve the resilience of cloud services, financial platforms, and AI training infrastructure that depend on datacenter fabrics. While evaluated on datacenter networks, the formulation extends naturally to other cyber-physical and distributed systems in which detection, diagnosis, and constrained recovery must be coordinated, including cloud-native platforms, edge networks, and critical infrastructure systems. By connecting these stages into one reproducible IEEE-style framework, this paper establishes a stronger foundation for future self-healing datacenter research. Reproducible artifacts including replay pipelines, orchestration scripts, and evaluation traces will be released.

References

[1] X. Chen, Y. Zhao, H. Liu, and J. Wang, "Predictive maintenance in datacenters: A machine learning approach," IEEE Transactions on Network and Service Management, vol. 18, no. 1, pp. 60–73, 2021.

TABLE IV
Summary of module-level results that support the unified self-healing architecture.

Module	Task	Key measured result	Operational significance
Failure prediction	Early warning on telemetry sequences	F1 0.3737 on zero-shot optical hard failures; F1 0.4677 on Cisco BGP failures within a 60 s warning window	Creates actionable lead time before full outage manifestation and supports proactive escalation.
Drift-adaptive IDS	Streaming attack and anomaly detection under transfer	Weighted F1 improves from 61.35% to 68.69% on full CICIDS2017 transfer without retraining; 98.05% weighted F1 on packet-capture replay; 0.2630 ms/flow latency	Maintains detection quality under distribution shift while remaining deployable at low latency.
Topology-aware RCA	Root-cause localization and temporal propagation tracking	Target-localization F1 0.8380, hidden-target accuracy 1.0000, temporal RCA accuracy 0.9394 with 5.2 s mean delay	Localizes faults even with incomplete telemetry and reconstructs causal chains for response.
Recovery engine	Safety-gated remediation with counterfactual validation	Reachability improves 0.9740 → 1.0000; recovery success 0.8182; unsafe actions blocked 100%	Converts diagnosis into measurable self-healing actions while suppressing harmful remediation.

[2] M. Roesch, "Snort: Lightweight intrusion detection for networks," in Proceedings of the 13th USENIX Conference on System Administration, 1999, pp. 229–238.

[3] 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, 2017, pp. 595–612.

[4] 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, 2021, pp. 131–146.

[5] S. Hochreiter and J. Schmidhuber, "Long short-term memory," Neural Computation, vol. 9, no. 8, pp. 1735–1780, 1997.

[6] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.

[7] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," ACM Computing Surveys, vol. 46, no. 4, pp. 44:1–44:37, 2014.

[8] 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, 2024, pp. 77–86.

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

[10] N. Moustafa and J. Slay, "Unsw-nb15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in 2015 Military Communications and Information Systems Conference, 2015, pp. 1–6.

[11] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in 2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications, 2009, pp. 1–6.

[12] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proceedings of the 4th International Conference on Information Systems Security and Privacy, 2018, pp. 108–116.

[13] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," in Proceedings of the 2007 SIAM International Conference on Data Mining, 2007, pp. 443–448.

[14] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," in Advances in Artificial Intelligence – SBIA 2004, 2004, pp. 286–295.

[15] E. S. Page, "Continuous inspection schemes," Biometrika, vol. 41, no. 1/2, pp. 100–115, 1954.

[16] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.

[17] L. Breiman, "Random forests," Machine Learning, vol. 45, no. 1, pp. 5–32, 2001.

[18] 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.

[19] 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.

[20] B. Lantz, B. Heller, and N. McKeown, "A network in a laptop: Rapid prototyping for software-defined networks," in Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks, 2010, pp. 19:1–19:6.

[21] Containerlab Team, "Containerlab," https://containerlab.dev/, 2026, documentation accessed March 18, 2026.

[22] A. Fogel, S. Fung, L. Pedrosa, M. Walraed-Sullivan, R. Govindan, R. Mahajan, and T. D. Millstein, "A general approach to network configuration analysis," in 12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15), 2015, pp. 469–483.