International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(21s) 2026 pp. 433-441
https://doi.org/10.70917/ijcisim-2026-5318
Article

Proactive Monitoring Using Streaming Telemetry for Enterprise Network Resilience

Miteshkumar Jagdishbhai Patel

Independent Researcher, USA

Abstract: Polling-based monitoring architectures impose a structural measurement ceiling on enterprise networks that hybrid infrastructure environments consistently exceed. Simple Network Management Protocol collection, operating at five-to-fifteen-minute intervals, cannot resolve the transient anomalies and short-duration control-plane events that cause a disproportionate share of service degradation in distributed manufacturing and corporate environments. This article argues that streaming telemetry, when architecturally integrated with machine learning-based anomaly detection, automated root cause analysis, and closed-loop remediation workflows, produces operational outcomes that polling-based models cannot approximate, and that the integration architecture connecting these components is the primary determinant of those outcomes. Analysis of enterprise hybrid deployment patterns suggests that telemetry pipeline design, adaptive baseline modeling, and remediation governance must be treated as interdependent architectural decisions rather than sequential technology adoptions. The resulting observability-driven operational framework substantially reduces mean time to detection and supports movement toward predictive and autonomous network operations at scale. This article is presented as an architectural and conceptual synthesis of the enterprise network observability literature rather than as an empirical study. It does not report pilot deployment measurements, controlled experiments, or benchmark data; its claims are intended to establish a research and practice agenda, and readers should evaluate them accordingly (see Section VII, Limitations).

Keywords: Streaming Telemetry, Enterprise Network Resilience, Observability, Anomaly Detection, Closed-Loop Automation, Root Cause Analysis, Network Operations, Hybrid Infrastructure, Machine Learning, Software-Defined Networking

1. Introduction

Measurement latency is the silent failure mode of polling-based enterprise monitoring, not dramatic enough to constitute an incident on its own, yet consequential enough to make a significant fraction of genuine incidents invisible until they have already degraded service. In manufacturing enterprise environments where network connectivity directly sustains production operations across dozens of geographically distributed sites, a short-lived routing instability event may complete its entire lifecycle within a single SNMP polling interval, leaving operations teams with degraded application performance records, zero network event records, and no actionable diagnostic path. The gap between event duration and measurement granularity is not a tuning problem with a configuration solution; it is an architectural property of polling-based collection that is incompatible with the temporal dynamics of modern hybrid networks [1].

What makes this limitation operationally acute is its compounding interaction with threshold-based alerting. Hybrid network environments, spanning on-premises data centers, cloud egress points, SD-WAN overlays, and site networks serving manufacturing plants and distribution centers, produce traffic patterns that vary substantially across operational windows, making static threshold calibration a persistent challenge [2]. Thresholds set conservatively to reduce false positives during peak periods miss genuine anomalies during baseline operations; thresholds set aggressively enough to catch gradual degradation patterns generate alert volumes that operators learn to deprioritise. The result is a monitoring posture that is simultaneously too sensitive and not sensitive enough, depending on time of day and operational phase.

Streaming telemetry offers a different architectural premise: rather than querying devices periodically, the network infrastructure itself exports operational state continuously, using gRPC transport and model-driven YANG schemas that describe interface counters, routing protocol state, queue depths, and control-plane events with a level of temporal precision that periodic SNMP polling cannot match [3], [16]. The data characteristics this produces, continuous temporal sequences rather than discrete periodic snapshots, make it possible for downstream analytics systems to observe the onset and progression of operational anomalies rather than merely their aftermath. That said, telemetry data without adequate downstream architecture produces a data volume problem rather than a visibility solution.

Four architectural layers determine whether a streaming telemetry deployment produces operational resilience or operational complexity: the pipeline infrastructure that manages data volume and ensures continuity, the machine learning models that establish adaptive behavioral baselines, the root cause analysis and alert correlation systems that compress diagnosis time, and the closed-loop remediation frameworks that convert high-confidence diagnoses into autonomous responses within defined guardrails. This article examines each layer and the integration decisions between them, arguing throughout that the architecture connecting these components, not the sophistication of any individual component, is where enterprise network resilience is built or degraded.

2. Limitations of Polling-Based Monitoring in Hybrid Enterprise Environments

Consider a hypothetical scenario, constructed here for illustration rather than drawn from a specific documented incident: a transient routing instability event affecting a manufacturing site's uplink lasts under two minutes, causes application session timeouts across the facility's production systems, and self-recovers before the next SNMP poll fires. The network management platform records no event. The application performance platform records latency spikes and session drops. The operators who investigate find no correlation between network and application records, declare the event unexplained, and close the incident, leaving the underlying condition free to recur. This is the predictable operational consequence of a measurement architecture whose polling interval can exceed the duration of a class of events that commonly affect distributed enterprise environments, and it reflects a structural limitation rather than an implementation deficiency [1].

Threshold-based alerting adds a second failure mode that interacts destructively with polling latency. The alert management challenge in enterprise environments serving manufacturing plants, distribution centers, and corporate campuses is not the absence of alerts but their abundance and low signal quality. Independent monitoring platforms covering network infrastructure, application performance, and synthetic transaction testing each generate their own alert streams against independently configured thresholds, producing alert volumes during complex incidents that can overwhelm triage capacity and desensitise operators to high-priority notifications [5]. Studies examining alert management in enterprise network operations consistently identify false positive rates in multi-platform monitoring environments as a primary contributor to extended incident response times, not because data is absent but because its volume and redundancy make extraction of actionable signal prohibitively expensive.

Hybrid infrastructure compounds both failure modes by adding architectural complexity that polling-based monitoring models were not designed to represent. East-west traffic within data center fabrics such as Cisco ACI generates endpoint mobility events, policy deployment confirmations, and fabric convergence signals at sub-second rates during normal operations, all of which produce operational states whose significance requires continuous observation to interpret correctly [6]. SD-WAN overlay behavior, where traffic path selection responds dynamically to circuit quality measurements, produces routing changes that are operationally intentional but can be indistinguishable from anomalous routing instability when viewed through five-minute polling snapshots. The monitoring model adequate for a static, single-vendor campus network is structurally misaligned with the operational reality of a hybrid enterprise environment today.

Taken together, these limitations establish a clear operational boundary. Polling-based monitoring and static threshold alerting represent a monitoring architecture that cannot be remediated through configuration improvements or additional alerting rules; the fundamental constraints of measurement latency, threshold calibration instability, and

multi-platform alert fragmentation are properties of the architecture itself. Enterprises that have outgrown this architecture require a different data collection model, a different analytics approach, and a different response framework. Streaming telemetry, machine learning-based detection, and closed-loop automation provide each of these in turn [7].

TABLE I. Comparison of Polling-Based vs. Streaming Telemetry Monitoring Architectures

Characteristic	SNMP Polling-Based	Streaming Telemetry
Data collection model	Periodic pull (5–15 min intervals)	Continuous push (sub-second intervals)
Transient event visibility	Events < polling interval invisible	All events captured in real time
Threshold sensitivity	Static, manually calibrated	Dynamic, ML-learned baselines
Multi-platform correlation	Manual, operator-dependent	Automated, topology-aware
Operational model supported	Reactive troubleshooting	Predictive / closed-loop automation

Note: this table summarises conceptual and qualitative distinctions drawn from the cited literature and the architectural analysis developed in this article. It is not a report of empirical benchmark measurements; comparative quantitative validation is identified as a priority for future work in Section VII.

3. Streaming Telemetry Pipeline Architecture for Enterprise-Scale Deployments

Before streaming telemetry can solve an analytics problem, it must solve a data engineering problem, and in enterprise environments, the data engineering challenge is substantial. Consider, for illustration, a moderately sized leaf-spine fabric (such as those built on Cisco Nexus 9000 switches) exporting gNMI telemetry at high frequency across interface, BGP, OSPF, and QoS data paths: the resulting ingest rates can exceed what naively designed pipelines can sustain during fabric convergence events, precisely the conditions under which continuous telemetry coverage matters most [8]. Pipeline architecture must therefore treat overload resilience as a first-class design requirement rather than an operational afterthought; a collector that falls behind and drops data during a major fabric event is operationally equivalent to no collector at all.

Subscription model selection at the collection layer carries architectural implications that extend through the entire pipeline. Dial-out subscriptions, where devices initiate telemetry export to configured collector endpoints, are operationally preferable in enterprise environments because they decouple data export from collector availability, allowing devices to continue streaming to local buffers during collector restart events. Multi-vendor environments introduce a normalisation requirement that single-vendor deployments avoid: YANG model paths for conceptually identical operational attributes differ across platforms and vendors [3]. Normalisation logic at the collection layer, mapping platform-specific schema paths to a unified canonical data model, is the engineering investment that makes multi-vendor telemetry analytically tractable.

Enrichment at the processing layer is the transformation that distinguishes an observability platform from a data lake. Raw telemetry counters, interface utilisation percentages, BGP prefix counts, queue drop rates, become operationally actionable when associated with the topology context that makes them interpretable: which application service relies on this interface, which sites are affected by this BGP peer's state, what is the redundancy status of the path this queue serves [6]. Stream processing frameworks that perform this enrichment inline, before data reaches the analytics layer, enable anomaly detection models to operate on enriched, contextualised records rather than raw device metrics.

Pipeline resilience design must confront the adversarial relationship between peak telemetry value and peak telemetry load. Graceful degradation mechanisms, adaptive sampling rate reduction triggered by ingestion queue depth thresholds, priority-based stream filtering that preserves control-plane state exports while shedding high-frequency interface counter updates, and geographically distributed collector deployment preventing single-region

failures from creating visibility gaps, provide the operational continuity that makes telemetry trustworthy as a monitoring foundation [9]. A pipeline designed for graceful degradation under load behaves predictably during the events that most challenge monitoring infrastructure, which is the operational property that makes it suitable as the data foundation for downstream automation systems.

TABLE II. Streaming Telemetry Pipeline: Architectural Layer Functions and Design Requirements

Pipeline Layer	Primary Function	Key Design Requirement	Failure Risk if Neglected
Collection	Device telemetry subscription and export	Multi-vendor YANG normalisation	Schema inconsistency in analytics
Transport	High-throughput message delivery	Overload resilience and backpressure	Data loss during peak events
Processing	Enrichment and context association	Topology-aware enrichment logic	Unactionable raw metrics
Analytics	ML-based anomaly detection	Adaptive baseline with concept drift handling	Threshold recalibration overhead

4. ML-Based Anomaly Detection: From Static Thresholds to Adaptive Baselines

Dynamic network environments defeat static threshold alerting not through any single spectacular failure but through the steady accumulation of miscalibration across thousands of monitored entities. Each threshold that is too high misses genuine degradation; each that is too low generates false alerts that consume operator attention and erode alert credibility over time. Machine learning-based anomaly detection sidesteps this calibration problem by learning the behavioral envelope of each monitored entity from its own operational history, adapting continuously as that history evolves, and flagging deviations from the learned envelope rather than from a human-assigned value [10]. In an enterprise environment where traffic patterns vary predictably by shift schedule, production cycle, and maintenance window, this adaptive baseline approach produces detection accuracy that static thresholds cannot approach without continuous manual recalibration.

Among unsupervised detection approaches applicable to enterprise network telemetry, isolation forests perform effectively on high-dimensional feature spaces where anomalies represent a small fraction of total observations, a characteristic that accurately describes enterprise network operations, where genuine anomalies are rare relative to normal operational data volume [11]. Autoencoders offer complementary capability for capturing complex multivariate relationships between correlated metrics; an autoencoder trained on normal operational data produces high reconstruction error for anomalous patterns, providing a detection signal that incorporates correlation structure between metrics rather than treating each independently. Graph-based fusion approaches extend this idea further by modeling explicit relationships between telemetry streams from different network elements, which can improve detection of anomalies that only manifest through cross-element correlation rather than in any single metric [18]. The operational requirement that often determines approach selection is interpretability: an anomaly detection system that produces a confidence score without explaining what triggered the detection places the diagnostic burden back on the operator.

Concept drift poses a sustained operational challenge that distinguishes production anomaly detection deployments from research evaluations. Following a major infrastructure change, an SD-WAN migration, a data center fabric extension, an ACI policy model restructuring, the behavioral envelopes that trained ML models learned no longer accurately represent normal operational behavior, causing detection accuracy to degrade progressively [10], [19]. This mirrors the domain-shift problem documented in intent-based data center assurance research, where models trained under one operational regime lose reliability once the underlying network intent or topology changes [24]. Continuous model performance monitoring, with automated retraining triggers that activate when detection accuracy metrics indicate distribution shift, is not an advanced feature of mature anomaly detection deployments; it is a basic

operational requirement without which detection accuracy degrades silently until the system produces more noise than signal.

The operational benefit of adaptive anomaly detection is best measured against a baseline that most enterprise environments can readily establish: the frequency and duration of transient events that cause application impact but produce no network event record in polling-based monitoring systems. Reducing mean time to detection for this class of previously invisible events represents a qualitatively different operational capability, not a marginal improvement in an existing metric but the creation of a new one [12]. The downstream value compounds through the diagnosis and response layers: events detected early carry more diagnostic context, support more accurate automated RCA, and admit a wider range of remediation actions before service impact becomes severe. The magnitude of this improvement will vary by environment and has not yet been established through controlled measurement; see Section VII.

TABLE III. ML Anomaly Detection Approaches: Enterprise Operational Characteristics

Approach	Strength	Enterprise Limitation	Primary Use Case
Isolation Forest	High-dimensional feature spaces	Limited temporal modeling	Interface utilisation anomalies
Autoencoder	Captures metric correlations	Requires training data volume	Multi-metric fabric health
DBSCAN Clustering	No labelled data required	Sensitive to density parameters	Routing topology anomalies
LSTM / RNN	Temporal pattern learning	Higher inference latency	Control-plane event sequences
Graph-based fusion	Models cross-element correlation	Higher modeling and data complexity	Multi-device correlated anomalies

5. Automated Root Cause Analysis and Alert Correlation

Multi-platform monitoring environments in enterprise operations produce a diagnosis bottleneck that neither better tools nor more experienced operators can fully resolve through manual effort alone. When a production incident in a hybrid environment triggers independent alert streams from network management, application performance monitoring, synthetic testing, and security event platforms simultaneously, the correlation work required to identify the causal event from among dozens of symptomatic alerts can consume substantial senior engineer time before any remediation action begins [13]. A significant share of this time is spent assembling evidence that automated systems are, in principle, well positioned to correlate far more quickly, a human attention tax that automated root cause analysis and alert correlation are specifically designed to reduce.

Topology-aware correlation engines approach the alert consolidation problem by treating the network topology graph as the primary organisational framework for alert relationships. When a failed interface on a WAN edge router generates cascading alerts from downstream switches, application servers, and synthetic monitoring probes, a correlation engine that maps each alert to its position in the topology graph can identify the common ancestor, the failed interface, and group all symptomatic alerts into a single correlated incident with the probable failure domain annotated [13]. This approach is conceptually aligned with recent work on assurance and conflict detection in intent-based networking, which similarly treats declared network intent and topology as the reference model against which observed state is checked [20]. Topology model accuracy is the operational dependency this approach introduces: stale IPAM records, undocumented connections, or CMDB data that does not reflect recent infrastructure changes produce correlation errors whose operational consequence, misdirected investigations, may be worse than no correlation.

Historical pattern recognition extends the correlation layer's capability beyond topology-based grouping to failure signature identification. Alert combination patterns associated with specific failure types, such as the sequence

of route withdrawal notifications, interface utilisation spikes, and application latency increases characteristic of a WAN circuit degradation event, can be learned from historical incident data and used to classify new incidents as they arrive, reducing the diagnosis phase from investigation to verification [14]. The data quality prerequisite for this capability, clean, consistently labelled historical incident records, represents a meaningful operational investment that enterprise teams must make deliberately.

Bringing topology-aware grouping and pattern-based classification together produces a diagnosis pipeline where automated systems assemble the evidence, identify the probable cause, and present a structured incident summary to the operator for confirmation. The operator's role shifts from correlator to validator, which preserves human judgment at the decision point that matters most, the transition from diagnosis to action, while reducing the evidence-assembly work that previously dominated incident response time [15].

6. Closed-Loop Remediation and the Transition to Autonomous Network Operations

The operational distinction between automation-assisted monitoring and closed-loop operations is precise: in the former, automation improves the speed and quality of information available to human decision-makers; in the latter, automation makes decisions and executes actions within defined boundaries for well-characterised failure patterns, with human oversight preserved at the point where action boundaries would be exceeded. This distinction parallels the broader discussion of bounded autonomy in agentic AI systems, where autonomous decision-making is deliberately scoped to well-characterised task classes rather than open-ended authority [21]. The gap between these two operational models is not primarily a technology gap but a design gap: the action library, guardrail framework, and feedback architecture that make autonomous execution safe and reliable require deliberate engineering that most telemetry deployments do not include.

Effective action library design begins with failure classification: each anomaly type that the detection and RCA layers can identify with high confidence corresponds to a class of remediation actions appropriate for that failure domain. Interface state resets, BGP session restoration sequences, traffic rerouting triggers via SD-WAN policy adjustment, and configuration rollback executions each address specific, well-understood failure patterns whose remediation steps are predictable, pre-testable, and reversible [15]. The operational risk of closed-loop execution is concentrated in the action selection logic, the mapping between identified failure type and remediation action, rather than in the execution mechanism itself.

Guardrail architecture addresses the risk of autonomous execution in environments where failures propagate across system boundaries in non-obvious ways. Scope limits define the maximum set of devices an autonomous action can affect; rollback conditions specify the post-execution telemetry criteria indicating an action produced an unintended outcome and trigger automatic reversal; escalation thresholds route scenarios that exceed the action library's coverage to human operators regardless of detection confidence [15]. A tiered execution model, fully autonomous for low-risk reversible actions, human-approved for moderate-impact operations, human-executed with automated diagnostic support for complex scenarios, distributes operational risk in proportion to action consequence, an approach consistent with zero-touch orchestration frameworks proposed for large-scale autonomous network management [17].

What closed-loop automation ultimately represents is the operational culmination of the integration architecture that streaming telemetry, ML anomaly detection, and automated RCA collectively enable. The value of any individual component in this chain is bounded by the weakest link in its connections: a high-fidelity telemetry pipeline feeding a poorly integrated analytics system produces noise; an accurate anomaly detection model connected to an imprecise action library produces operational risk; a sophisticated correlation engine operating on incomplete topology data produces misdirected investigations. Organisations that approach observability as an architectural design problem, rather than a series of independent technology adoptions, are better positioned to build the integration quality that turns individual capabilities into compounding operational value.

TABLE IV. Closed-Loop Remediation: Tiered Execution Model

Tier	Action Type	Confidence Required	Human Role	Example Actions
Tier 1	Fully autonomous	High confidence, reversible	Post-action review	Interface reset, BGP session restore
Tier 2	Automated + approval	High confidence, moderate impact	Pre-execution approval	Traffic rerouting, config rollback
Tier 3	Human-executed	Any confidence level	Full execution authority	Complex multi-domain remediation

7. Limitations, Risks, and Adoption Barriers

The architectural framework developed in this article carries costs and risks that a purely capability-focused discussion can obscure. First, continuous streaming telemetry and adaptive ML-based detection are not free: sustaining sub-second collection, enrichment, and inference at enterprise scale requires ongoing compute, storage, and network capacity, and the retraining cycles needed to keep behavioral baselines current impose a recurring operational and financial cost that grows with the size and heterogeneity of the environment. Enterprises evaluating this architecture should budget for this as a standing operational expense rather than a one-time implementation cost.

Second, closed-loop remediation expands the attack surface of network operations in ways that deserve explicit attention. A system that acts autonomously on telemetry-derived conclusions creates an incentive for an adversary to manipulate the telemetry itself, for example by spoofing or degrading signals in ways designed to trigger an unwanted automated action, or to suppress signals that would otherwise trigger a legitimate one. Authenticating telemetry sources, constraining the scope and reversibility of autonomous actions, and monitoring the action layer itself for anomalous behavior are necessary complements to the detection and remediation capabilities described above, not optional hardening steps.

Third, the organizational and change-management barriers to adopting this architecture are substantial and easy to underestimate. Operations teams accustomed to manual diagnosis and manual change control may reasonably resist ceding action authority to an automated system, particularly for actions with production impact. Building the trust required for tiers 2 and 3 of the execution model described in Section VI typically requires a phased rollout, starting with low-risk, easily reversible Tier 1 actions and expanding scope only as the system demonstrates reliable behavior, along with investment in staff training on the new diagnostic and governance workflows.

Finally, and most fundamentally, this article presents an architectural and conceptual analysis rather than an empirical evaluation. The claimed benefits, faster detection of previously invisible events, reduced diagnosis time, and safe autonomous remediation, are argued from first principles and from the individual capabilities documented in the cited literature, but they have not been validated here through a pilot deployment, a controlled experiment, or measured before-and-after performance data in a live enterprise environment. Readers should treat the framework as a design hypothesis and an agenda for future validation rather than as an evaluated result.

8. Conclusion

The structural inadequacy of polling-based monitoring in hybrid enterprise environments is not a problem that can be resolved by improving tools within the existing paradigm. Streaming telemetry, machine learning-based anomaly detection, automated root cause analysis, and closed-loop remediation together define an alternative architecture, one that aligns the measurement model with the temporal dynamics of modern distributed networks, replaces brittle manual threshold calibration with adaptive baselines, compresses the evidence-assembly phase of incident diagnosis, and converts high-confidence diagnoses into governed autonomous action. The analysis developed across Sections II through VI argues that the integration quality between these four layers, not the sophistication of

any single layer, is what determines whether a deployment produces genuine operational resilience or simply a more complex monitoring stack.

That argument, however, is presented here as a conceptual and architectural synthesis, and Section VII sets out why it should be read that way: the computational and financial cost of continuous adaptive monitoring, the expanded attack surface introduced by autonomous remediation, and the organizational trust-building required for staged autonomy are real costs that any adoption decision must weigh against the anticipated benefits, and none of the benefits claimed here have yet been demonstrated through controlled measurement in a production environment.

A useful research and practice agenda follows directly from this gap. The most immediate priority is empirical validation: a pilot deployment or controlled study that measures mean time to detection, mean time to resolution, and false-positive rates before and after adoption of the four-layer architecture would test the central claims of this article directly. A second direction is security-focused: characterising and hardening closed-loop systems against telemetry manipulation and action-layer abuse is an open problem that becomes more pressing as remediation scope expands. A third direction connects this architecture to network digital twins, which model network behavior in a virtual replica that could be used to test anomaly detection models and remediation actions safely before they are granted authority over production infrastructure, an integration that recent digital twin research for network operations and management suggests is both feasible and increasingly practical [22], [23]. Finally, the organizational question, how enterprises build sufficient trust to expand autonomous action scope over time, is as much a subject for future study as any of the technical layers discussed here.

REFERENCES

1. Xu Chen, "Toward Automated Network Management and Operations," ResearchGate, 2012. Available: https://www.researchgate.net/publication/48927835_Toward_Automated_Network_Management_and_Operations
2. A. Leivadeas and M. Falkner, "A Survey on Intent-Based Networking," IEEE Communications Surveys & Tutorials, vol. 25, no. 1, pp. 625–655, 2022. Available: https://ieeexplore.ieee.org/document/9925251
3. R. Boutaba et al., "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," Journal of Internet Services and Applications, vol. 9, no. 1, pp. 1–99, 2018. Available: https://www.researchgate.net/publication/325107577_A_comprehensive_survey_on_machine_learning_for_networking_evolution_applications_and_research_opportunities
4. M. Bjorklund et al., "RFC 8342: Network Management Datastore Architecture (NMDA)," RFC Editor, United States, 2018. Available: https://dl.acm.org/doi/10.17487/RFC8342
5. P. Almasan et al., "Network Digital Twin: Context, Enabling Technologies, and Opportunities," IEEE Communications Magazine, vol. 60, no. 11, pp. 22–27, 2022. Available: https://ieeexplore.ieee.org/document/9795043
6. E. Coronado et al., "Zero Touch Management: A Survey of Network Automation Solutions for 5G and 6G Networks," IEEE Communications Surveys & Tutorials, vol. 24, no. 4, pp. 2535–2578, 2022. Available: https://ieeexplore.ieee.org/document/9913206
7. A. Sivanathan, H. H. Gharakheili, and V. Sivaraman, "Managing IoT Cyber-Security Using Programmable Telemetry and Machine Learning," IEEE Transactions on Network and Service Management, vol. 17, no. 1, pp. 60–74, 2020. Available: https://ieeexplore.ieee.org/document/8981946
8. Wenxin Zhong and Xianhong Zhao, "Research on DCI Streaming Telemetry System Based on gNMI," 4th International Conference on Information Science, Parallel and Distributed Systems (ISPDS), 2023, pp. 1–8. Available: https://ieeexplore.ieee.org/document/10235592
9. Francesco Paolucci et al., "Network Telemetry Streaming Services in SDN-Based Disaggregated Optical Networks," Journal of Lightwave Technology, vol. 10, no. 1, pp. A58–A67, 2018. Available: https://ieeexplore.ieee.org/document/8263153
10. D. Kong et al., "Toward Security-Enhanced In-Band Network Telemetry in Programmable Networks," IEEE Transactions on Network and Service Management, vol. 20, no. 1, pp. 137–155, 2024. Available: https://ieeexplore.ieee.org/document/10764753
11. Piotr Lechowicz et al., "Trade-Offs in Implementing Unsupervised Anomaly Detection with TAPI-Based Streaming Telemetry," IEEE 25th International Conference on High Performance Switching and Routing (HPSR), 2024, pp. 1–6. Available: https://ieeexplore.ieee.org/document/10635922
12. Vignesh Karunakaran et al., "TAPI-based Telemetry Streaming in Multi-domain Optical Transport Network," Optical Fiber Communications Conference and Exhibition (OFC), 2024, pp. 1–3. Available: https://ieeexplore.ieee.org/document/10526531

13. Sukhyun Nam et al., "Network Anomaly Detection Based on In-band Network Telemetry with RNN," IEEE International Conference on Consumer Electronics - Asia (ICCE-Asia), 2020, pp. 1–6. Available: https://ieeexplore.ieee.org/document/9276768
14. L. Yang et al., "Enabling AutoML for Zero-Touch Network Security: Use-Case Driven Analysis," IEEE Transactions on Network and Service Management, vol. 21, no. 3, pp. 3555–3582, 2024. Available: https://ieeexplore.ieee.org/document/10472316
15. S. Rani et al., "Network Slicing for Zero-Touch Networks: A Top-Notch Technology," IEEE Network, vol. 37, no. 5, pp. 16–24, 2023. Available: https://ieeexplore.ieee.org/document/10274564
16. Selector, "Network Telemetry in 2025: How It Works, Protocols and Use Cases," 2025. Available: https://www.selector.ai/learning-center/network-telemetry-how-it-works-protocols-and-use-cases/
17. Hatim Chergui et al., "Toward zero-touch management and orchestration of massive deployment of network slices in 6G," IEEE Wireless Communications, vol. 29, no. 1, pp. 86–93, 2022. Available: https://www.eurecom.fr/en/publication/6858
18. Yunfeng Duan et al., "MFGAD-INT: in-band network telemetry data-driven anomaly detection using multi-feature fusion graph deep learning," Journal of Cloud Computing, vol. 12, no. 1, 2023. Available: https://link.springer.com/article/10.1186/s13677-023-00492-w
19. Vincenzo Agate et al., "Anomaly Detection for Reoccurring Concept Drift in Smart Environments," 18th International Conference on Mobility, Sensing and Networking (MSN), 2022, pp. 113–120. Available: https://ieeexplore.ieee.org/document/10076623
20. Molka Gharbaoui et al., "Assurance and Conflict Detection in Intent-Based Networking: A Comprehensive Survey and Insights on Standards and Open-Source Tools," IEEE Transactions on Network and Service Management, 2026. Available: https://ieeexplore.ieee.org/document/11334180
21. Deepak Bhaskar Acharya, Karthigeyan Kuppan, and B. Divya, "Agentic AI: Autonomous Intelligence for Complex Goals—A Comprehensive Survey," IEEE Access, vol. 13, pp. 18912–18936, 2025. Available: https://ieeexplore.ieee.org/abstract/document/10849561
22. Daniel González-Sánchez et al., "Toward Building a Digital Twin for Network Operations and Management," IEEE Open Journal of the Communications Society, vol. 6, pp. 2583–2598, 2025. Available: https://ieeexplore.ieee.org/document/10919072
23. Baolin Qin et al., "Machine and Deep Learning for Digital Twin Networks: A Survey," IEEE Internet of Things Journal, vol. 26, no. 3, pp. 1529–1573, 2024. Available: https://ieeexplore.ieee.org/document/10570372
24. Steve Lévesque et al., "Network Assurance in Intent Based Data Center Networking: A Domain Shift Approach," 16th International Conference on Network of the Future (NoF), 2025, pp. 1–6. Available: https://ieeexplore.ieee.org/document/11223335