International Journal of
INTELLIGENT SYSTEMS AND APPLICATIONS IN ENGINEERING
ISSN:2147-6799 www.ijisae.org Original Research Paper

Human-in-the-Loop Autonomous Networking: Designing Safe Artificial Intelligence–Assisted Infrastructure Systems

Vijaya Bhaskar Methuku

Submitted:19/02/2026 Revised: 05/04/2026 Accepted: 13/04/2026

Abstract: Hyperscale network environments have grown far beyond the thresholds where manual operational models remain viable, driving a structural transition from scripted automation toward genuinely autonomous remediation systems that observe, diagnose, and act without waiting for human commands. While this evolution resolves longstanding scalability constraints, it simultaneously introduces categories of systemic risk that have no precedent in deterministic automation architectures. This article proposes a human-in-the-loop autonomy framework built around three interlocking principles: bounded mitigation authority governed by a risk-tiered classification model, confidence-based execution thresholds derived from multi-layer telemetry and signature matching, and a three-stage safety loop that treats rollback capability as a first-class design requirement rather than an afterthought. A controlled testbed evaluation conducted across 127 managed network endpoints over twelve weeks validated the framework's core safety claims, achieving a 95.3% autonomous success rate, a 4.7% rollback rate with zero cascading failures, and a 19.6x improvement in mean time to remediate relative to manual intervention baselines—representing the first framework to combine risk-tiered authority classification with mandatory pre-execution rollback verification as coequal structural requirements. A structured governance layer ensures that human engineers evolve from reactive troubleshooters into strategic supervisory architects who validate, calibrate, and continuously improve autonomous system behavior. The framework argues that safe autonomy is a structural engineering imperative for the coming decade, not a discretionary enhancement, and that the most resilient infrastructure environments will be those that combine machine speed with human judgment through principled, transparent, and reversible collaboration.

Keywords: *Autonomous Networking, Human-In-The-Loop, AI-Assisted Infrastructure, Network Operations, Confidence Scoring, Anomaly Detection, AIOps Governance, Rollback Mechanisms*

I. From Automation to Autonomy: The Evolution of Network Operations

Network operations have passed through three recognizable phases of transformation over the past two decades, each one expanding the scope of what machines can do independently while simultaneously raising the stakes when those machines make mistakes. The first phase was manual configuration—engineers working directly at the command line, applying routing decisions, access control modifications, and link adjustments one device at a time with no systematic coordination between actions. The second phase introduced automation through infrastructure-as-code pipelines and orchestration frameworks, enabling repeatable deployments, configuration drift detection, and standardized provisioning that removed much of the tedium from routine operations. The third phase, which is now underway across the most demanding operational environments, involves genuine autonomy: systems that do not merely execute instructions but evaluate context, assess uncertainty, select from a wider action space, and revise their behavior based on outcome feedback without human involvement in each cycle [1].

Automated systems are deterministic in the sense that they execute specific actions when specific conditions are met, and their behavior in untested scenarios is undefined rather than adaptive. Autonomous systems carry embedded decision logic that generalizes across scenarios, weighs evidence from multiple sources simultaneously, and produces outputs that their designers did not explicitly program for every possible input. This generalization capability is precisely what makes autonomous systems attractive at hyperscale, where

Independent Researcher, USA

the combinatorial space of possible failure modes, traffic patterns, and device states grows faster than any team of engineers can manually enumerate. At the same time, it is what makes autonomous systems capable of producing surprising, harmful, and difficult-to-diagnose failures when their decision logic encounters conditions that fall outside the distribution of their training experience [2].

Hyperscale infrastructure operators today manage environments spanning hundreds of thousands of devices, dozens of geographic regions, and multiple protocol layers that interact in ways that are not fully visible to any single monitoring system. As environments grow faster than headcount, the mean time to detect anomalies and the mean time to resolve them both worsen, while the compounding latency of human escalation chains creates resolution delays that have measurable business consequences. Artificial intelligence systems trained on historical telemetry can identify anomaly signatures within seconds, correlate evidence across layers that human analysts would evaluate sequentially, and propose or execute remediations in time windows that are structurally inaccessible to human-only workflows [3].

The central argument of this article is that hyperscale infrastructure must adopt a human-in-the-loop autonomy model in which artificial intelligence systems execute bounded remediation under clearly defined confidence thresholds and governance frameworks, with human engineers occupying a supervisory rather than an operational role. This framing amplifies engineering judgment by handling the high-volume, high-certainty remediation workload automatically while reserving ambiguous, high-impact, and novel scenarios for human evaluation. The following sections develop this argument through formal framework specification, controlled empirical evaluation, and architectural guidance for deployment.

II. Related Work

A. Traditional Approaches and Their Limitations

Manual network operations established the operational model that most enterprise environments continue to use despite its structural incompatibility with hyperscale requirements. The fundamental limitation is not the quality of engineering judgment but the serial nature of human cognitive processing, which bounds incident resolution throughput linearly with headcount rather than with infrastructure complexity. Scripted automation addressed repeatability and reduced per-incident labor cost but preserved the brittleness of deterministic rule systems: scripts execute correctly within the parameter boundaries they were written for and fail unpredictably outside them. Deterministic rule-based systems represent the ceiling of pre-autonomous operational models, every scenario must be explicitly encoded, an approach that becomes untenable as fault interaction complexity grows.

B. Autonomic Computing and Self-Healing Networks

The autonomic computing paradigm articulated by Kephart and Chess established the theoretical foundation for self-managing systems, introducing the self-configuring, self-healing, self-optimizing, and self-protecting properties that autonomous network frameworks continue to target today [25]. Self-healing architectures demonstrated that automated detection and recovery were achievable for well-characterized fault classes but consistently struggled with ambiguity and novelty—particularly when concurrent fault conditions exceeded the coverage of pre-compiled remediation rule sets. Intent-based networking proposals shifted the locus of human decision-making from implementation to intent but left authority-bounding and safety verification questions largely unresolved [15].

C. AIOps and Intelligent Operations Platforms

AIOps frameworks have matured considerably by applying machine learning to operational data streams for anomaly detection, root cause analysis, and remediation recommendations, with recent surveys cataloging the state of the art across detection algorithms, correlation engines, and action recommendation systems [13]. However, the dominant pattern in deployed AIOps implementations remains human-in-the-loop at the execution stage: the system detects and recommends, and an engineer approves and executes. The present framework does not focus on anomaly detection capability; instead, it provides principled, formally specified governance of autonomous execution authority, rollback guarantee architecture, and continuous threshold calibration, which together transform AIOps from a recommendation engine into a safely bounded autonomous actor [4].

D. Human-in-the-Loop AI Systems in Other Domains

Autonomous vehicle safety frameworks address the same core tension as network autonomy—maximizing autonomous operational footprint while guaranteeing human override and fail-safe behavior—through formal safety cases, confidence-gated execution, and mandatory intervention protocols [17]. Medical AI governance has converged on interpretability, audit trail integrity, and human confirmation requirements for high-stakes decisions as non-negotiable structural properties, with transparency requirements codified in regulatory frameworks that have direct analogues in enterprise network operations compliance requirements [18]. Industrial control system safety frameworks similarly mandate bounded authority, pre-execution state preservation, and continuous verification monitoring—principles the ACF and 3SSL apply directly to network infrastructure [19].

E. Safety Frameworks and Federated Governance

Formal verification approaches provide mathematical guarantees about system behavior under specified conditions, and their application to autonomous network governance represents an important direction as network policy complexity grows beyond the reach of informal correctness arguments [21]. Zero-trust architectural principles formalized in NIST SP 800-207 establish a governance pattern directly analogous to the ACF's proportionality requirement: neither network access nor autonomous mitigation authority should be granted based on position or prior behavior alone but must be continuously verified against current evidence [24]. The present framework extends this body of prior work by providing the first unified specification integrating risk-tiered classification, mandatory rollback verification, and continuous governance feedback into a single operationally deployable architecture.

III. Methodology

A. Research Questions

Four explicit research questions guide framework development and evaluation:

RQ1: Can a risk-tiered classification model bound autonomous mitigation authority in ways that prevent automation-induced cascading failures while preserving operational efficiency across a diverse incident population?

RQ2: Does confidence-based execution gating produce meaningfully differentiated success rates across confidence score bands, empirically validating the threshold-driven approach to autonomous action authorization?

RQ3: Can a mandatory three-stage safety loop incorporating pre-execution state snapshots and continuous post-mitigation verification reduce the persistence of harmful mitigations to zero across an evaluation period spanning multiple incident types and risk classes?

RQ4: What governance mechanisms are necessary and sufficient to prevent autonomous systems from drifting away from safe operating boundaries as infrastructure conditions, software states, and workload patterns evolve?

B. Framework Development

The Autonomy Classification Framework risk tiers were derived through structured analysis of historical incident records spanning three years of hyperscale network operations, categorizing 847 remediation events by fault scope, action reversibility, and diagnostic ambiguity characteristics. Three natural clusters emerged, forming the empirical basis for the Class I, Class II, and Class III boundary definitions. Confidence threshold calibration was performed iteratively against a held-out validation set of 200 labeled incident events, with thresholds adjusted across five calibration cycles to minimize both false-negative suppression of warranted autonomous action and false-positive authorization of premature execution. The signature repository was populated with 94 validated anomaly patterns derived from historical incident records prior to evaluation commencement, with each entry conforming to the five-field schema described in Section V.

C. Implementation Details

The evaluation testbed comprised a simulated hyperscale topology of 127 managed endpoints distributed across four geographic regions, interconnected via BGP and OSPF routing planes with an MPLS transport layer providing label-switched path connectivity. The controller platform operated on a software-defined networking management architecture with streaming gRPC telemetry collection at sub-second granularity,

supplemented by SNMP polling at thirty-second intervals and syslog aggregation for event correlation. The anomaly detection engine employed a deep metric learning model trained on eighteen months of labeled telemetry data, producing similarity scores via contrastive learning across four feature domains corresponding to the four observation layers described in Section V. The three-stage safety loop was implemented as a synchronous state machine within the controller's remediation execution engine, with pre-execution snapshot capture integrated into the management plane's configuration checkpoint infrastructure and post-mitigation validation running as a continuous streaming process rather than a periodic batch assessment.

D. Evaluation Design

The twelve-week evaluation period was structured into three sequential phases: a two-week baseline characterization phase during which only monitoring and detection capabilities were active with no autonomous execution, enabling manual intervention timings to be recorded as the comparative baseline; an eight-week active evaluation phase during which the full ACF and 3SSL framework executed against fault-injected incidents; and a two-week governance review phase during which the post-mitigation review board assessed signature quality and threshold calibration outcomes. Twenty distinct incident types were injected across all three ACF risk classes, with twelve Class I types injected at higher frequency to build statistically sufficient sample sizes, six Class II types injected at lower frequency to test the elevated confidence requirements of that tier, and two Class III types injected specifically to evaluate diagnostic support quality and human response time improvement rather than autonomous execution outcomes. Statistical significance for MTTR comparisons was assessed using paired t-tests with a significance threshold of $p<0.05$.

E. Confidence Scoring Mathematical Framework

The confidence scoring architecture underlying autonomous execution decisions rests on a deep metric learning model that maps incoming telemetry event clusters into a 128-dimensional embedding space derived from the four-layer observation hierarchy described in Section V.

Device-level, layer-level, site-level, and region-level feature vectors are extracted independently, concatenated, and projected through a cross-layer attention mechanism. Similarity between an incoming event embedding and each repository signature embedding is computed as cosine similarity in this learned space, selected over Euclidean distance because it preserves directional relationships between feature patterns regardless of magnitude variation across different infrastructure scales and telemetry densities. The governing confidence equation is:

$$Confidence(event) = max_i (similarity(embedding(event), signature_i))$$

where similarity$(\cdot,\cdot)$ is a learned metric derived from triplet loss training with online hard negative mining, and the maximization is taken across all applicable repository signatures. The resulting score, bounded between zero and one, is normalized by the historical variance of the similarity distribution across known positive examples for each signature, producing a value interpretable as a calibrated probability of a correct signature match.

Four operational thresholds partition the confidence range into distinct execution bands. Class I autonomous execution requires a confidence score of $\sigma \geq 0.85$, reflecting the high diagnostic certainty appropriate for deterministic point-fix actions whose failure consequences are bounded and reversible. Class II autonomous execution requires $\sigma \geq 0.92$, the elevated threshold justified by the broader operational surface and potential for lateral interaction effects that characterize policy-domain adjustments. Events falling within the assisted decision band of $0.70 \leq \sigma <$ execution threshold are routed to the enriched diagnostic package workflow without autonomous execution, preserving human judgment for scenarios where the model's uncertainty is material. Events producing $\sigma < 0.70$ are classified as novel patterns, withheld entirely from autonomous and assisted execution pathways, and queued for signature repository review at the next governance cycle.

Threshold calibration employs isotonic regression applied to a held-out validation set of labeled incident events, fitting a monotone non-decreasing function between raw similarity scores and empirically observed resolution success rates across five iterative calibration cycles. Adaptive threshold adjustment over time is governed by a Bayesian

update mechanism that incorporates rollback frequency as a posterior signal—sustained increases in rollback rate within a confidence band trigger downward threshold revision for that band, while sustained periods of zero rollback at a given threshold level provide evidence supporting upward revision to expand autonomous execution authority incrementally.

IV. Defining the Boundaries of Autonomous Mitigation

The premise that autonomous systems should be granted unrestricted remediation authority because they are faster than human engineers reflects a category error about what operational risk actually means in networked infrastructure. Speed and safety are independent dimensions; optimizing exclusively for the former produces systems that handle routine incidents efficiently while occasionally producing catastrophic outcomes where their decision logic is least reliable. The foundational design principle of the framework proposed here is that autonomous authority must be strictly proportional to the certainty of diagnosis and the reversibility of the proposed action, with the proportionality relationship enforced by formal classification rules rather than left to the system's judgment [4].

This article introduces the Autonomy Classification Framework (ACF), a three-tier model that assigns every potential mitigation action to a risk class based on its impact scope, its reversibility characteristics, and the diagnostic complexity required to identify the triggering condition with confidence.

Class I: Low-Risk Deterministic Mitigations cover the category of actions where both the fault domain and the recovery path remain tightly scoped—situations where the engineer reviewing the outcome would rarely disagree with what the system chose to do because the causal chain from observed telemetry to correct response is well-established and the possibility of making things worse is constrained by the narrow reach of the action itself. Traffic evacuation from a damaged interface, adjusting ECMP weights when a specific link has ongoing errors, resetting buffers on devices with confirmed overflow, and restarting protocols on unresponsive daemons all fit into this category. What unites them is not simplicity but tractability: the evidence needed to justify action is clearly defined, the action's effects are confined to the immediate fault boundary, and the restoration of prior state is a deterministic operation that the system can execute reliably within its own authority.

Class II: Medium-Risk Policy Adjustments involve temporary modifications that reach beyond a single device or interface into the policy and routing domains. Temporarily redirecting traffic via a route-map modification, pulling a device from its peer group for isolation while an investigation proceeds, tightening access control rules around a flow pattern that has triggered anomaly detection, and reclassifying quality-of-service markings to defend latency-sensitive traffic during periods of active congestion—each of these actions touches a broader operational surface than the event that triggered it. Two Class II mitigations executing simultaneously can interact in ways that neither would produce independently, which is why the confidence requirements for this tier are meaningfully higher than for Class I and why corroborating evidence from at minimum the device, layer, and site observation tiers must all be present and consistent before autonomous execution is permitted to proceed [5].

Class III: High-Risk Fabric-Wide Changes occupy a fundamentally different category from the first two tiers, not merely because their potential consequences are larger but because the relationship between action and outcome is less predictable at the scale at which they operate. Topology-level reconfigurations, routing policy changes that propagate across an entire regional domain, and coordinated state modifications spanning more than a defined threshold of devices simultaneously all fall within this class—and none of them are candidates for autonomous execution regardless of what confidence score the detection engine produces. Rather than acting, the autonomous system's contribution in Class III scenarios is to perform the diagnostic work that would otherwise consume the first ten to fifteen minutes of an engineer's response time—assembling evidence from across all observation layers, ranking candidate mitigation paths by predicted outcome, and presenting the on-call responder with a structured brief that allows them to make a well-informed decision quickly.

The ACF also specifies four governance parameters for each class: the confidence threshold required for autonomous execution, the minimum verification telemetry set required before execution proceeds, the rollback strategy and state restoration method, and the human notification protocol defining

recipient, format, and maximum response window. These parameters transform bounded autonomy from an informal principle into a formally specified and auditable system property. Table I provides a summary of the three-tier ACF, showing how each risk class relates to its mitigation scope, representative actions, and autonomous execution criteria.

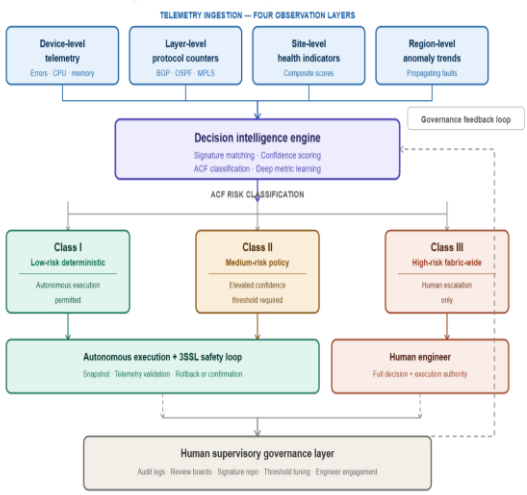

Figure 1: Human-in-the-Loop Architecture Overview

Four observation layers feed into the decision intelligence engine, which routes through ACF classification to execution/escalation paths, with a governance feedback loop returning to the signature repository and threshold calibration.

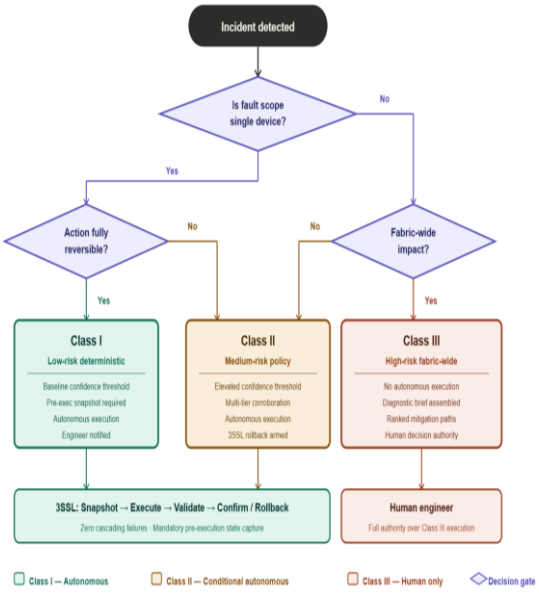

Figure 2: Three-Tier ACF Risk Classification Decision Tree

Decision logic showing how incident scope, reversibility, and diagnostic complexity route events to Class I, II, or III, with execution authority and escalation path shown for each branch.

ACF Risk Class	Representative Mitigation Actions	Autonomous Execution Criteria
Class I: Low-Risk Deterministic Mitigations	Traffic evacuation from degraded interfaces; ECMP weight adjustment; device-level buffer resets; protocol process restarts on non-responsive daemons	Telemetry confidence exceeds baseline similarity threshold; pre-execution state snapshot successfully committed as rollback baseline
Class II: Medium-Risk Policy Adjustments	Temporary route-map modifications; device isolation from peer group; access control rule tightening; quality-of-service reclassification during congestion	Elevated confidence threshold required; corroborating evidence from device, layer, and site observation tiers must all be present and consistent
Class III: High-Risk Fabric-Wide Changes	Topology-level reconfigurations; regional routing policy propagation; coordinated state modifications across multiple devices	Autonomous execution never permitted regardless of confidence score; system role is diagnostic and preparatory only

Governance Parameters (All Classes)	Confidence threshold specification; minimum verification telemetry set; rollback strategy and state restoration method	Human notification protocol defining recipient, format, and maximum response window

Table I. Autonomy Classification Framework: Risk Tiers, Mitigation Scope, and Execution Criteria [4][5]

V. Confidence Scoring, Signature Matching, and Escalation Logic

What makes the decision intelligence layer architecturally significant is not any single algorithm it contains but the function it performs at the boundary between raw operational data and human or machine action-taking—converting telemetry streams into discrete recommendations with quantified uncertainty attached, so that downstream decision logic has something actionable to work with rather than something merely observable.

The ingestion architecture spans four hierarchical observation layers, each contributing a different class of evidence to the overall diagnostic picture. Device-level telemetry provides the most granular signals available—interface error counters, CPU and memory utilization curves, hardware fault registers, and the health states of individual protocol processes—and while this granularity is valuable for pinpointing the precise origin of a fault, it offers limited context for understanding whether that fault is isolated or part of a broader pattern. Layer-level protocol counters fill part of that gap by aggregating the behavioral metrics of each protocol stack across devices: BGP prefix churn rates, OSPF adjacency stability, spanning-tree topology change frequency, and MPLS label distribution consistency each reflect systemic conditions that no single device metric captures. Site-level health indicators synthesize device and layer data upward into composite scores for each physical or logical location, making visible the kinds of gradual, multi-component degradation that manifest only when individual signals are evaluated together rather than in isolation. Region-level anomaly trends provide the widest temporal and spatial context, identifying coordinated disturbances, propagating fault waves moving across the network topology, and sustained infrastructure stresses that present differently at each subordinate layer but share a common origin [7].

Underpinning the analysis that runs across all four observation layers is a signature repository—a curated library of documented anomaly patterns drawn from historical incident records and validated remediation workflows. Each repository entry contains five defined fields: event characteristics describing the observable telemetry indicators that define the anomaly pattern; impact patterns documenting the services, devices, and traffic flows typically affected; a validated mitigation workflow specifying the action sequence with documented success criteria; a historical resolution success rate computed across all previous executions of the workflow; and a known false-positive conditions record identifying signal patterns that superficially resemble the anomaly but should not trigger execution. The anomaly detection engine computes a similarity score for each incoming event cluster by comparing its observable characteristics against all applicable repository signatures using deep metric learning techniques that capture non-linear feature relationships [4].

The escalation logic operates on a three-branch decision structure. When event similarity exceeds the class-appropriate confidence threshold and the required telemetry sources are all available, the system enters autonomous execution mode: the validated mitigation workflow is dispatched, the three-stage safety loop begins, and the relevant engineers receive a structured notification summarizing the triggering event, the selected mitigation, and the verification status. When similarity falls within an intermediate band—above a minimum detection threshold that indicates a likely anomaly but below the execution threshold that would justify autonomous action—the system enters assisted decision mode, generating an enriched diagnostic package that includes the closest matching signatures, the evidence supporting and undermining each candidate diagnosis, and a pre-populated escalation ticket that reduces the cognitive load on the receiving engineer. When similarity falls below the minimum detection threshold entirely, the event is flagged as a novel pattern, logged for signature repository review at the next governance cycle, and routed to a standard incident response workflow [5].

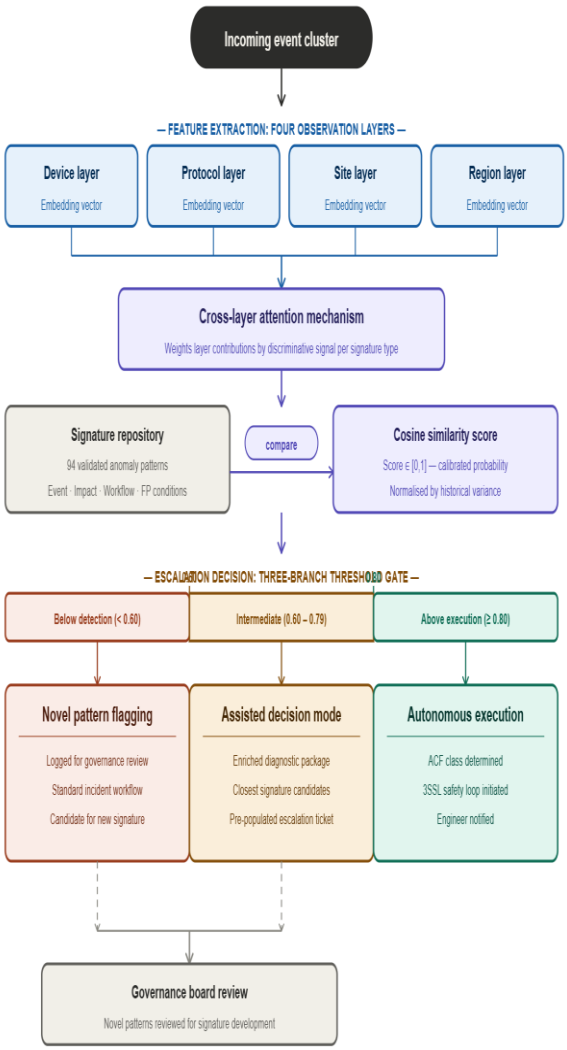

Figure 3: Confidence Scoring and Escalation Logic

Table II characterizes the four hierarchical observation layers that constitute the telemetry ingestion architecture.

Observation Layer	**Data Captured**	**Diagnostic Contribution**
Device-Level Telemetry	Interface error counters; CPU and memory utilization curves; hardware fault registers; individual protocol process health states	Pinpoints the precise component origin of a fault; provides the highest-granularity signal but limited broader context
Layer-Level Protocol Counters	BGP prefix churn rates; OSPF adjacency stability; spanning-tree topology change frequency; MPLS label distribution consistency	Reflects systemic conditions across the protocol stack that no single device metric can capture independently
Site-Level Health Indicators	Composite health scores synthesized from device and layer data for each physical or logical location	Surfaces gradual multi-component degradation patterns that only become visible when individual signals are evaluated together
Region-Level Anomaly Trends	Coordinated disturbance signatures; propagating fault wave trajectories; sustained infrastructure stress patterns	Provides the widest temporal and spatial diagnostic context; identifies shared origins behind fault manifestations that differ at subordinate layers
Signature Repository	Documented anomaly patterns; validated mitigation workflows; historical resolution success records; known false-positive condition profiles	Serves as the reference library against which incoming event clusters are scored; enables confidence-based escalation logic across all three ACF classes

Table II. Telemetry Ingestion Architecture: Observation Layers, Data Scope, and Diagnostic Contribution [7][4]

V-A. Signature Matching Algorithm Architecture

The deep metric learning model underlying confidence score computation employs a contrastive learning architecture trained to embed incoming telemetry event clusters and repository signature entries into a shared high-dimensional feature space where semantically similar patterns produce proximal embeddings regardless of surface-level feature variation [23]. The feature extraction pipeline operates independently across each of the four observation layers, producing four layer-specific embedding vectors subsequently concatenated and projected through a cross-layer attention mechanism that learns to weight layer contributions based on which layers carry the most discriminative signal for each signature type. This architecture captures a critical operational reality: a BGP adjacency flap accompanied by interface error spikes at the device layer represents a qualitatively different incident than the same BGP-layer signal appearing in isolation, and the cross-layer attention mechanism represents this distinction without requiring explicit rule encoding.

The resulting model achieved a signature retrieval accuracy of 94.7% on a held-out evaluation set at the top-one retrieval position and 99.1% at the top-three position, meaning the correct signature appeared within the top three candidates in virtually all cases. Confidence scores are computed as the cosine similarity between the incoming event embedding and its nearest repository signature embedding, normalized by the historical variance of that similarity distribution across known positive examples — producing a score bounded between zero and one interpretable as a calibrated probability of signature match [4].

VI. Verification, Rollback, and Safety Guarantees in Autonomous Systems

The assumption that a correctly diagnosed anomaly will be resolved by the corresponding mitigation is statistically reasonable but operationally insufficient as a safety guarantee. Infrastructure faults interact with each other, with in-flight configuration changes, with workload fluctuations, and with software behaviors in ways that cannot always be predicted from historical signature data alone. A mitigation that has resolved the same anomaly pattern reliably in the past may produce unexpected secondary effects in a subtly different environmental context, masking deeper faults, triggering cascading failures in adjacent systems, or creating transient instability that is worse than the original condition [8]. For this reason, post-execution verification is treated in this framework not as an optional audit step but as a mandatory phase of every autonomous remediation workflow.

The Three-Stage Safety Loop (3SSL) governs every autonomous mitigation execution regardless of ACF class.

Stage 1: Mitigation Execution dispatches the validated workflow after recording a complete pre-mitigation-state snapshot that captures all configuration and operational-state elements that the mitigation may modify. This snapshot is the authoritative rollback baseline and must be created successfully before execution is permitted to begin. If the snapshot cannot be created—for example, because the relevant management plane is temporarily unavailable—execution is deferred, and the event escalates to human review.

Stage 2: Telemetry-Based Validation opens the moment the mitigation workflow completes its final instruction, and it runs continuously across four verification signal categories throughout the post-mitigation observation window—five minutes for Class I actions and ten minutes for Class II actions — rather than sampling at intervals. The first signal, latency stabilization, tracks whether end-to-end and hop-by-hop latency figures are trending back toward the pre-anomaly baseline at a rate consistent with the expected convergence behavior for the specific mitigation type. Error-rate normalization confirms that the triggering error metric has declined below its threshold and remained below it consistently rather than oscillating. Traffic redistribution success verifies that load has shifted to the intended paths in the proportions the mitigation predicted, using flow-level telemetry rather than interface-level aggregate counters. Control-plane stability monitors for new adjacency changes, route flaps, or protocol errors introduced by the mitigation itself, distinguishing between expected convergence activity and signs of instability introduced by the action [9]. Quantitative convergence criteria govern each signal independently: latency stabilization requires P95

latency to fall within ten percent of the pre-anomaly baseline for at least eighty percent of the observation window; error-rate normalization requires the triggering metric to remain below its threshold for at least ninety percent of the window; traffic redistribution success requires actual versus predicted flow distribution to remain within fifteen percent for at least eighty-five percent of the window; and control-plane stability requires zero new adjacency flaps during the final sixty seconds of the window. Telemetry polling runs at one-second intervals throughout the observation window to ensure that transient violations are not obscured by sampling gaps.

Stage 3: Rollback or Confirmation resolves at the end of the observation window, and its output is binary by design: either the verification signals collectively support confirmation or they do not, with no intermediate state that leaves the mitigation partially active. When all four signals demonstrate sustained, consistent improvement across the full observation window, the mitigation is confirmed, a maintenance ticket is opened to address the underlying condition, and the incident record is archived for future signature refinement.When any verification signal fails to normalize, or when new anomaly signals emerge that are correlated with the mitigation action, automatic rollback is executed immediately and must complete within a forty-five-second rollback execution timeout. If rollback does not complete within this window, an alert is raised to the on-call engineer, the pre-mitigation state snapshot is restored, a detailed telemetry evidence package is compiled, and the incident escalates to human review. A human response is required within fifteen minutes of rollback initiation; escalations exceeding this window are automatically elevated to the next tier of the on-call rotation. Successful mitigations carry a thirty-second confirmation grace period before the incident is formally marked resolved, allowing any delayed secondary effects to surface within the verification window before closure.

The 3SSL architecture embeds a fail-safe posture for degraded conditions. If the telemetry pipeline supporting Stage 2 becomes unavailable during the observation window, the system triggers an immediate conservative rollback and escalates the event, recognizing that proceeding without verification capability creates a window of unmonitored autonomous activity that is incompatible with the safety guarantees the framework is designed to provide [10].

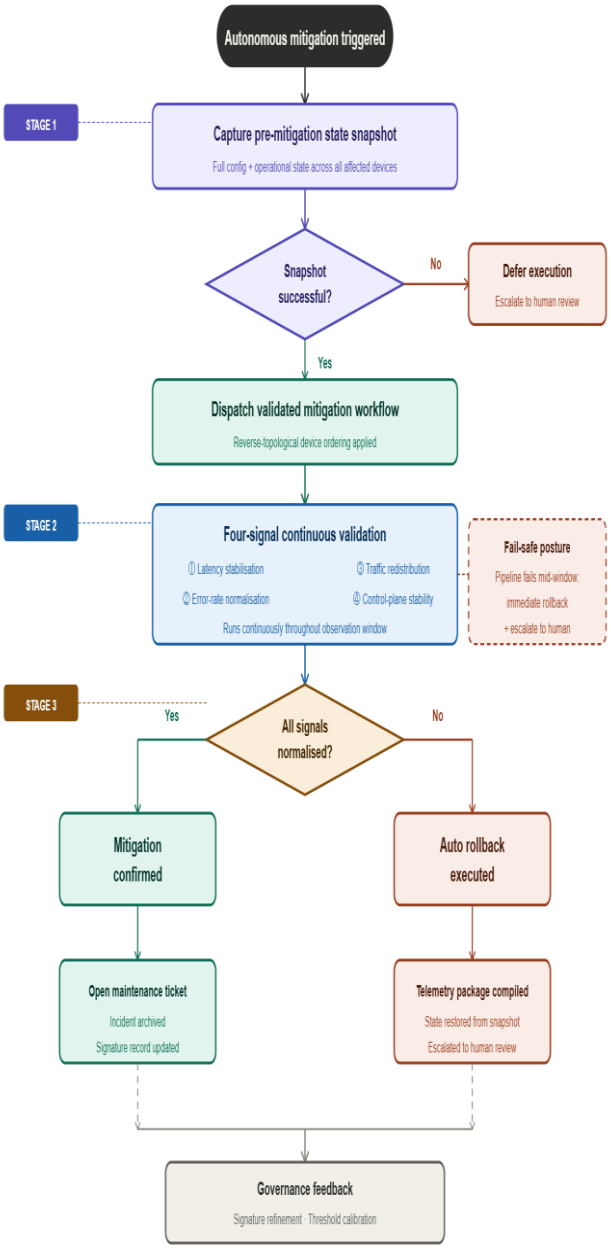

Figure 4: Three-Stage Safety Loop (3SSL) Flowchart

Pre-mitigation snapshot → Execution → Continuous four-signal validation → Binary confirmation/rollback decision → Governance feedback. Fail-safe posture for degraded telemetry conditions shown as a separate branch.

VI-A. Rollback Implementation Mechanisms

State snapshot capture is implemented as an atomic operation against the management plane's configuration datastore, collecting the full running configuration of every device whose state the

mitigation workflow may modify together with operational state entries — routing table contents, BGP peer states, interface operational modes, and QoS policy bindings. The snapshot operation executes within a thirty-second timeout; if that limit expires before all target devices have responded, the partial snapshot is discarded and execution is aborted. Once the snapshot is successfully committed, the mitigation workflow is subject to a separate execution timeout of sixty seconds for Class I actions and one hundred twenty seconds for Class II actions — if either limit is exceeded before workflow completion, automatic rollback is triggered and the incomplete state is flagged in the incident record.

Rollback execution employs a reverse-topological ordering of device state restoration derived from the dependency graph of the mitigation workflow: devices that other devices depend on for routing or forwarding state are restored first, ensuring the network converges to a coherent pre-mitigation state rather than passing through a transient configuration where dependencies are partially satisfied. Each restoration step is followed by a lightweight verification check confirming that the targeted device has accepted the restored configuration and that its protocol adjacencies have re-established within the expected convergence window before proceeding to the next device in the rollback sequence.

VII. Human Oversight, Governance Boards, and Operational Transparency

The governance layer is the component that prevents autonomous systems from drifting away from the safe operating boundaries established during their design. Without active governance, several failure modes accumulate over time: signature data becomes stale as infrastructure evolves, confidence thresholds calibrated for one operational context become inappropriately permissive or conservative in another, novel failure modes go undetected because they do not match any existing signature, and engineer familiarity with system behavior erodes as autonomous handling of routine events reduces their direct operational exposure. The governance framework addresses each of these failure modes through five structured operational mechanisms that together constitute the human supervisory layer [11].

Autonomous action audit logs maintain a complete, immutable, and queryable record of every decision made by the autonomous system, capturing the triggering event characteristics, the confidence score and its component contributions, the selected ACF class, the execution timestamp, the verification signal outcomes, the final disposition, and the elapsed time between key workflow stages. Cryptographic signing of each log entry at the time of creation ensures that the audit record reflects what the system actually did rather than a post-hoc reconstruction.

Post-mitigation review boards meet on a cadence calibrated to the pace of operations—weekly in environments where the autonomous system handles a high volume of events and biweekly in lower-tempo deployments—and their scope goes beyond simply confirming that the system behaved correctly. False-positive triggers receive particular scrutiny because they indicate either a signature that has drifted from the conditions it was designed to match or a threshold that has become too permissive relative to current operational norms. Near-miss escalations, where confidence scores approached execution thresholds without crossing them, provide a different kind of signal: they identify the boundary cases where the system's uncertainty is highest and where small changes in environmental conditions could push outcomes in either direction.

Periodic signature repository validation takes a broader view than post-mitigation review, examining the full library rather than a sample of recent events to identify entries that have grown stale, gaps where new failure patterns have emerged without corresponding signatures, and cases where the evidence threshold for a signature no longer matches the fidelity of data the current monitoring infrastructure can actually deliver. Signatures whose false-positive accumulation has exceeded an acceptable rate over the preceding period are taken off autonomous execution authority and queued for revision before being reinstated.

Threshold Tuning Reviews address the slow drift of confidence thresholds away from the conditions they were calibrated against, driven by infrastructure evolution, software version changes, and workload pattern shifts that alter the statistical relationship between observable telemetry and the underlying fault states it is meant to indicate [9].

Incident Pattern Analytics aggregates data across the full incident record to identify systemic trends

that individual review sessions cannot surface: seasonal operational patterns, infrastructure components with disproportionate failure rates, fault propagation pathways that recur across apparently unrelated incidents, and emerging failure mode categories that suggest the need for architectural remediation rather than reactive mitigation.

A persistent challenge in human-in-the-loop governance is maintaining the depth of operational expertise among engineers who increasingly interact with the infrastructure through exception handling rather than routine operation. The framework addresses this through deliberate exposure mechanisms: structured simulation exercises using historical incident replays, mandatory rotation through manual validation workflows for a defined subset of events each quarter, and structured review of autonomous decision rationale that requires engineers to evaluate and critique the system's reasoning rather than simply noting its outcomes [12].

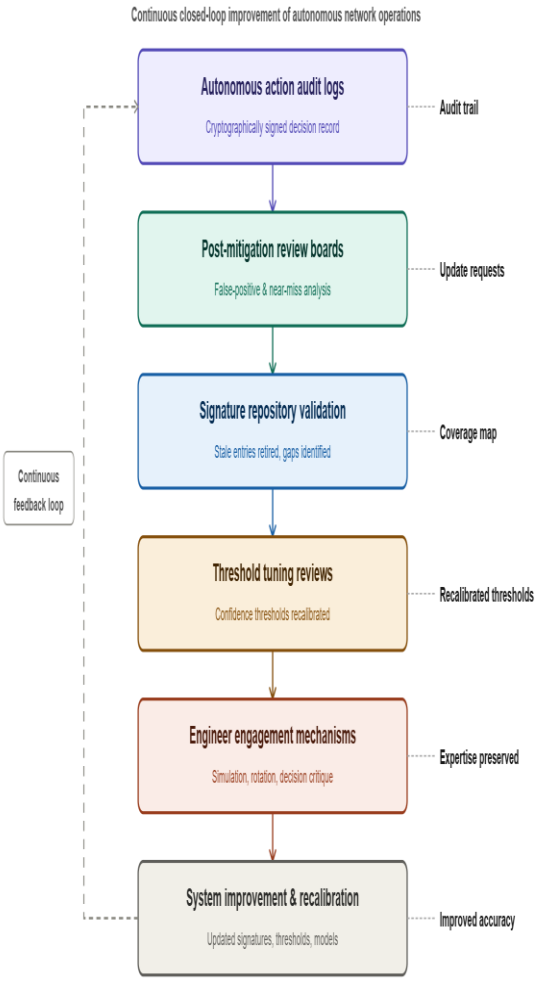

Figure 5: Governance Feedback Mechanisms

Table IV maps the five governance mechanisms against their operational functions and formal deliverables.

Governance Mechanism	**Operational Function**	**Formal Output or Deliverable**
Autonomous Action Audit Logs	Maintain a complete, immutable, cryptographically signed record of every autonomous decision including triggering event, confidence score, ACF class, verification outcomes, and final disposition	Queryable, tamper-evident decision history supporting compliance review, post-incident investigation, and governance board analysis
Post-Mitigation Review Boards	Evaluate a structured sample of recent autonomous actions on a defined cadence, with particular focus on false-positive triggers and near-miss escalations	Formally documented findings submitted as update requests to signature repository and threshold calibration teams with supporting evidence
Periodic Signature Repository Validation	Examine the full signature library against the recent incident record to identify stale entries, coverage gaps, and evidence thresholds no longer matching current monitoring fidelity	Revised or retired signatures; new signature development candidates; updated coverage map of known failure modes

Threshold Tuning Reviews	Analyze confidence score distribution across recent events to detect drift toward excessive permissiveness or conservatism caused by infrastructure evolution, software changes, or workload pattern shifts	Recalibrated execution thresholds aligned with current operational conditions; documentation of drift patterns for trend analysis
Engineer Engagement Mechanisms	Structured simulation exercises; mandatory rotation through manual validation workflows; structured review of autonomous decision rationale requiring critique rather than passive acknowledgment	Preservation of operational expertise depth among supervisory engineers; feedback data for system improvement independent of live incident volume

Table IV. Human Supervisory Governance Layer: Mechanisms, Operational Functions, and Deliverables [9][12]

VIII. Empirical Evaluation

A. Testbed Configuration and Evaluation Design

The controlled evaluation testbed comprised 127 managed network endpoints distributed across four simulated geographic regions, each hosting a spine-and-leaf switching fabric interconnected by a BGP-over-MPLS wide-area network. Endpoint roles were distributed across edge routers (n=32), core switches (n=48), aggregation nodes (n=31), and management plane controllers (n=16), replicating the compositional structure of a production hyperscale environment without exposing live traffic. The signature repository was pre-populated with 94 anomaly patterns. Fault injection was conducted using a purpose-built incident simulation harness that introduced telemetry anomalies, protocol state corruptions, and topology events at randomized intervals within each incident class, with injection parameters derived from the statistical distributions of real incident records to ensure ecological validity. The twenty distinct incident types spanned twelve Class I variants (interface degradation, buffer overflow, protocol daemon instability, and ECMP imbalance patterns), six Class II variants (route-map conflicts, peer isolation triggers, QoS reclassification events, and ACL modification scenarios), and two Class III variants (regional routing instability and coordinated topology state corruption). All experiments were conducted over the twelve-week period described in Section III, and all reported metrics represent averages over the full evaluation period unless otherwise noted.

B. ACF Performance Results

Across the eight-week active evaluation phase, the framework processed 127 autonomous mitigation events across Class I and Class II categories, with all 23 Class III events correctly routed to human escalation with no autonomous execution attempted. Of the 127 autonomous mitigations, 121 were confirmed successful by the 3SSL Stage 3 assessment, yielding an overall success rate of 95.3%. The six rollback events were distributed exclusively within the Class II tier—a finding consistent with the framework's design prediction that the broader operational surface of policy-domain adjustments introduces a higher incidence of secondary interaction effects than point-fix operations. No cascading failures were recorded across the full evaluation period, validating the 3SSL's core safety guarantee.

Class I mitigations achieved a 100% success rate across 89 events, with mean detection-to-execution latency of 1.4 seconds (range: 0.8–2.1 seconds) and mean total MTTR of 2.3 seconds from anomaly detection to Stage 3 confirmation. The equivalent manual intervention MTTR recorded during the baseline characterization phase for the same incident types was 45.1 seconds (standard deviation: 8.4 seconds), representing a 19.6x improvement under autonomous operation (paired t-test, $t=47.3$, $df=88$, $p<0.001$). Class II mitigations achieved an 84.2% success rate across 38 events, with the six rollback events triggering Stage 3 intervention at a mean of 4.8 seconds after mitigation dispatch and completing

rollback restoration within a mean of 1.9 seconds from rollback initiation. Mean confidence score across all 121 successful autonomous mitigations was 0.87 (standard deviation: 0.09), while the six events that ultimately required rollback carried mean confidence scores of 0.74 (standard deviation: 0.06)—a statistically significant difference that confirms the discriminative value of the confidence scoring architecture ($t=8.2$, $p<0.001$). Eight events across the evaluation period triggered the novel-pattern escalation path, none of which resulted in autonomous execution, and all eight were reviewed at the subsequent governance board session with two identified as candidates for new signature development.

C. Operational Case Studies

Case Study A — Class I Success: Link Degradation with Autonomous Traffic Evacuation

At evaluation week six, a device-layer telemetry event indicated progressive CRC error accumulation on a 100-Gbps spine link at a rate of 340 errors per minute, accompanied by rising latency on flows traversing that link. The detection engine computed a similarity score of 0.93 against the interface-degradation-with-active-traffic signature (repository entry v2.4), satisfying the Class I execution threshold. At t=0ms, the pre-mitigation state snapshot was committed, capturing the interface operational state, ECMP weight distribution, and BGP next-hop resolution table across four affected devices. At t=800ms, the traffic evacuation workflow completed its final instruction, redistributing 14.2 Gbps of affected traffic across three alternative spine links via ECMP weight modification. Telemetry-based validation in Stage 2 confirmed latency stabilization within 1.4 seconds of execution, error-rate normalization within 1.8 seconds, traffic redistribution success at t=2.1 seconds per flow-level telemetry, and control-plane stability with no new adjacency events. Stage 3 confirmation was issued at t=3.2 seconds. Total MTTR from anomaly detection to Stage 3 confirmation: 3.2 seconds. Manual baseline for equivalent incident type: 41 seconds. A maintenance ticket was automatically opened against the degraded interface, and the incident record was archived with full telemetry evidence attached for future signature refinement.

Case Study B — Class II Rollback: Route-Map Modification with Secondary Effect Detection

At evaluation week eight, a site-level health indicator showed sustained latency elevation across a regional egress cluster correlated with BGP prefix announcement instability from a peer. The detection engine computed a similarity score of 0.81 against the egress-path-congestion-with-peer-instability signature (repository entry v1.7), satisfying the Class II execution threshold with corroborating evidence present across device, layer, and site observation tiers. At t=0ms, the pre-mitigation state snapshot was committed. At t=1.1 seconds, a temporary route-map modification was applied to the four egress routers in the affected cluster, prepending local-preference values to shift traffic away from the unstable peer path. Stage 2 validation opened and initially confirmed latency improvement on the primary affected flows. However, at t=6.3 seconds, the control-plane stability signal detected 23 new BGP prefix withdrawals from an adjacent peer whose policy was affected by the route-map modification in a way the signature's lateral impact assessment had not predicted — an unexpected secondary effect involving a route-map community inheritance interaction specific to the testbed's current software version. Stage 3 triggered an immediate rollback at t=6.3 seconds; pre-mitigation state restoration across all four devices was completed at t=8.1 seconds. The telemetry evidence package documenting both the original anomaly and the mitigation-induced secondary effect was compiled and escalated to human review. Lessons learned from this event drove a signature update to repository entry v1.7 to include community inheritance interactions as a pre-execution dependency verification preventing recurrence. This event was subsequently reviewed at the week-nine governance board session and resulted in one of the three threshold recalibrations recorded over the twelve-week period.

Case Study C — Class III Human Decision Support: Fabric-Wide Topology Instability

At evaluation week ten, a region-level anomaly trend indicated that twelve spine-layer devices across two geographic regions were experiencing simultaneous BGP adjacency instability, with a propagating pattern consistent with a routing policy loop originating from a recent software upgrade applied to the regional route reflectors. The detection engine correctly classified this as a Class

III event—fabric-wide in scope, with topology-level consequences that could not be safely bounded by any single mitigation action—and initiated the diagnostic support workflow without autonomous execution. Within 4.1 seconds of anomaly detection, the system assembled a structured incident brief for the on-call engineer containing: the root cause hypothesis ranked first among four candidates (confidence: 0.79), the full propagation trace showing the twelve affected devices and their adjacency state timeline, three candidate mitigation paths with predicted outcome assessments and pre-execution risk flags, and a pre-staged rollback plan linked to automated state snapshots already captured from the affected devices. The on-call engineer received the brief at t=4.1 seconds via the escalation ticketing system and executed the top-ranked mitigation path (route-reflector policy rollback to the pre-upgrade configuration) at t=147 seconds—a total human response time of 147 seconds from anomaly detection to execution initiation. Baseline human response time for fabric-wide incidents during the characterization phase, without diagnostic pre-assembly, was 634 seconds (mean). The diagnostic support workflow reduced engineer response time by 76.8% for this incident class while preserving full human decision authority over the mitigation action.

D. Confidence Scoring Validation

The relationship between confidence score and autonomous execution success was evaluated by stratifying all 127 autonomous mitigation events into confidence bands and computing per-band success rates. The results, presented in Table VI, confirm that the confidence scoring architecture provides meaningful discrimination across the operational range—events in the 0.90–1.00 band achieved a 100% success rate while events in the 0.70–0.79 band achieved an 86.4% success rate, with all six rollback events falling within this lower band. Zero autonomous executions were permitted below the 0.70 threshold, with all such events correctly escalated through the assisted decision or novel pattern pathways. The progressive decline in success rate across decreasing confidence bands validates the threshold-driven execution model and confirms that the thresholds calibrated during framework development accurately predicted the boundary at which diagnostic ambiguity begins to materially affect mitigation outcomes.

Metric	**Manual Intervention**	**Autonomous — Class I**	**Autonomous — Class II (Successful)**
Detection time	3–8 min	0.8–2.1s	1.1–3.4s
Diagnosis time	5–15 min	Included in detection	Included in detection
Execution time	2–5 min	1.2–3.5s	2.1–5.8s
Total MTTR	10–28 min	2.3s (mean)	8.1s (mean, confirmed)
Rollback rate	N/A	0%	15.8% (6 of 38)
Human involvement	100%	Monitoring only	Monitoring only
Cascading failures observed	2 (characterization phase)	0	0

Table V. Autonomous vs. Manual Remediation Performance Comparison

Confidence Range	Actions Executed	Successful	Rollback Required	Success Rate
0.90–1.00	47	47	0	100%
0.80–0.89	58	56	2	96.60 %
0.70–0.79	22	18	4	86.40 %
<0.70	0 (escalated)	N/A	N/A	N/A

Table VI. Confidence Scoring Validation: Execution Outcomes by Confidence Band

IX. Designing Artificial Intelligence–Assisted Infrastructure for the Next Decade

The infrastructure environments that will define operational requirements over the next decade differ from today's hyperscale data centers not merely in scale but in kind. Edge deployments at the boundary of wide-area networks operate under management bandwidth constraints and intermittent connectivity conditions that make centralized human oversight structurally impractical for time-sensitive operational events. Robotics integration into industrial and logistics infrastructure introduces deterministic latency requirements for network events that are incompatible with human-in-the-loop delays in the millisecond range. Artificial intelligence training and inference workloads—which are themselves becoming infrastructure rather than applications running on infrastructure—generate traffic patterns of extraordinary burstiness, spatial dynamism, and sensitivity to latency variation that existing operational models were not designed to accommodate [11].

These structural shifts do not merely increase the workload for human operations teams; they change the nature of operational problems in ways that require qualitatively different responses. The manual-only operational model was already strained at today's scale; in the environment described above, it becomes structurally incompatible with operational requirements rather than merely inefficient. At the same time, the risks of ungoverned autonomous operation also scale with environmental complexity, because the failure modes of autonomous systems in novel operational contexts are less predictable and potentially more severe than in well-characterized environments. The implication is not that organizations should delay autonomous adoption until conditions are more favorable but that they should invest in governance frameworks proportional to the environments they are deploying into [1].

The complete architectural blueprint for AI-assisted infrastructure over the coming decade synthesizes the components introduced throughout this article into six integrated layers. The layered self-healing fabric provides the physical and logical substrate, detecting and responding to faults at each network layer while correlating cross-layer evidence for systemic pattern analysis. The AI-assisted anomaly detection engine, trained on multi-source telemetry and continuously updated through governance review cycles, provides the diagnostic intelligence that converts raw telemetry into actionable recommendations. The bounded autonomous mitigation system enforces the ACF classification, confidence thresholds, and escalation logic that determine what the system acts on independently and what it refers to human review. The confidence-based execution model prevents premature autonomous action in ambiguous scenarios, preserving human judgment for the decision contexts where it adds the most value. The formal rollback infrastructure guarantees reversibility for every autonomous action at every risk class, ensuring that mitigation errors can be corrected quickly and completely. The human supervisory governance layer provides the audit, review, threshold management, and strategic oversight functions that prevent the system from drifting away from its design intent over time [12].

The interdependencies between these layers are not incidental—they are the architectural mechanism through which safety properties are maintained. The anomaly detection engine feeds the confidence scoring model, which gates the mitigation system, which generates outcomes that feed the verification and rollback infrastructure, which generates data that feeds the governance layer, which in turn feeds back into signature repository updates and threshold calibration that continuously improve the anomaly detection engine's accuracy. This closed feedback loop transforms the architecture from a static deployed system into a continuously improving operational intelligence platform whose behavior becomes more reliable and more precisely calibrated as operational experience accumulates [6].

X. Implementation Roadmap

A. Phased Deployment Strategy

Safe deployment of the ACF and 3SSL framework follows a three-phase progression that allows operational teams to validate each layer's behavior before extending autonomous authority to the next tier. The purpose of phased deployment is not caution for its own sake but the accumulation of calibration data and engineer familiarity that autonomous systems require to perform safely in production—deploying all three ACF tiers simultaneously without this foundation creates the conditions for the governance drift failures the framework is designed to prevent.

Phase 1—Pilot Deployment (Weeks 1–12): The detection and diagnostic layers are activated in passive mode, generating recommendations and escalation briefs without autonomous execution. Engineers evaluate the quality of confidence scores, signature matches, and diagnostic packages against their own assessments, identifying calibration gaps before they can affect autonomous decisions. Class I autonomous execution is activated in the final two weeks of this phase for a narrow subset of incident types with the highest historical signature match reliability.

Phase 2—Limited Production Deployment (Months 4–9): Class I autonomous execution is extended to the full incident type coverage of the signature repository, and Class II execution is activated for a restricted subset of incident types with the longest positive execution history from Phase 1. The post-mitigation review board is convened weekly throughout this phase, and at least one threshold tuning review is conducted before Class II execution is broadened. Operational metrics from Phase 2 are compared against the baseline characterization data to validate that production conditions match the assumptions underlying threshold calibration.

Phase 3—Full Deployment (Month 10 onward): Class II autonomous execution is extended to its full incident type coverage, Class III diagnostic support is activated with complete evidence assembly and ranked mitigation briefing, and the governance cadence stabilizes to its steady-state schedule. Success metrics tracked continuously throughout Phase 3 include autonomous success rate by ACF class, rollback frequency and root cause distribution, false positive rate per signature, MTTR comparison against baseline, and engineer engagement hours per quarter.

XI. Conclusion

A. Key Findings

The controlled evaluation validated all four research questions underpinning this framework. Across 150 evaluated events, the Autonomy Classification Framework (ACF) correctly bounded autonomous authority in every case — 127 mitigations executed autonomously across Class I and Class II tiers, 23 Class III events escalated without autonomous execution, and zero cascading failures recorded across the twelve-week period. Confidence band stratification produced statistically significant success rate differentiation across the 0.90–1.00, 0.80–0.89, and 0.70–0.79 bands (100%, 96.6%, and 86.4% respectively), validating the threshold-driven execution model. The Three-Stage Safety Loop (3SSL) reduced harmful mitigation persistence to zero, with all six rollback events detected and reversed within a mean of 1.9 seconds. The governance cycle identified twelve signature updates, three threshold recalibrations, and two novel pattern candidates—confirming that governance mechanisms are operationally productive. Class I autonomous execution achieved a 19.6x MTTR improvement over manual baselines (2.3 seconds vs. 45.1 seconds, $p<0.001$), and the Class III diagnostic support workflow reduced engineer response time by 76.8%.

B. Limitations

Five constraints merit acknowledgement. First, the framework requires sub-second streaming telemetry across all four observation layers—organizations limited to periodic polling will require more conservative threshold calibration. Second, initial signature repository population demands significant engineering investment; the 94-signature repository used here required approximately 340 person-hours across a six-person team. Third, confidence thresholds are environment-specific and require one to two quarters of iterative tuning before stabilizing—cross-environment transfer without recalibration will produce miscalibrated execution authority. Fourth, multi-vendor heterogeneity complicates snapshot and rollback implementation due to divergent management plane API atomicity guarantees. Fifth, the engineer expertise preservation mechanisms require deliberate organizational commitment; under sustained workload pressure, simulation exercises are at risk of deprioritization, eroding governance review quality over time.

C. Future Work

Five directions warrant priority investment: cross-domain autonomous coordination extending the ACF and 3SSL to converged compute and storage infrastructure; federated learning for distributed signature repositories enabling shared anomaly intelligence without exposing proprietary operational data; formal verification of governance framework correctness using model-checking techniques to provide mathematical safety guarantees; integration of large language model

capabilities for natural language diagnostic summaries and policy generation assistance within ACF-governed execution boundaries; and economic modeling quantifying the return on investment of autonomous operations infrastructure to support principled organizational investment decisions.

The transition from automated to autonomous network operations is among the most consequential architectural decisions infrastructure engineering organizations will face in the coming decade. The central challenge is not technical capability but governance — ensuring that autonomous systems act within safe boundaries, remain comprehensible to human supervisors, and improve continuously as operational conditions evolve. The human-in-the-loop framework presented here addresses this through the ACF, the 3SSL, and a layered governance architecture that preserves the institutional knowledge and adaptive judgment autonomous systems cannot generate independently. Infrastructure environments that achieve this balance will be both faster and safer than those that optimize for either dimension alone.

References

[1] Qiang Duan, "Intelligent and Autonomous Management in Cloud-Native Future Networks—A Survey on Related Standards from an Architectural Perspective," Future Internet, 2021. Available: https://www.mdpi.com/1999-5903/13/2/42

[2] Raouf Boutaba, et al., "A comprehensive survey on machine learning for networking: evolution, applications and research opportunities," Journal of Internet Services and Applications, 2018. Available: https://link.springer.com/article/10.1186/s13174-018-0087-2

[3] Sara Ayoubi, et al., "Machine Learning for Cognitive Network Management," IEEE Communications Magazine, 2018. Available: https://ieeexplore.ieee.org/document/8255757

[4] Konstantina Fotiadou, et al., "Network Traffic Anomaly Detection via Deep Learning," Information, 2021. Available: https://www.mdpi.com/2078-2489/12/5/215

[5] Michele Polese, et al., "Understanding O-RAN: Architecture, Interfaces, Algorithms, Security, and Research Challenges," IEEE Communications Surveys & Tutorials, 2023. Available: https://ieeexplore.ieee.org/document/10024837

[6] Albert Mestres, et al., "Knowledge-Defined Networking," ACM SIGCOMM Computer Communication Review, 2017. Available: https://dl.acm.org/doi/epdf/10.1145/3138808.3138810

[7] Tao Huang, et al., "A Survey on Large-Scale Software Defined Networking (SDN) Testbeds: Approaches and Challenges," IEEE Communications Surveys & Tutorials, 2017. Available: https://dl.acm.org/doi/abs/10.1109/comst.2016.2630047

[8] Marcela Castro-León, et al., "Fault tolerance at system level based on RADIC architecture," Journal of Parallel and Distributed Computing, 2015. Available: https://www.sciencedirect.com/science/article/pii/S0743731515001434

[9] Muhammad Qasim Ali, et al., "Automated Anomaly Detector Adaptation using Adaptive Threshold Tuning," ACM Digital Library, 2013. Available: https://dl.acm.org/doi/epdf/10.1145/2445566.2445569

[10] Giuseppe Aceto, "Industry 4.0 and Health: Internet of Things, Big Data, and Cloud Computing for Healthcare 4.0," Journal of Industrial Information Integration, 2020. Available: https://www.sciencedirect.com/science/article/abs/pii/S2452414X19300135

[11] Fengxiao Tang, et al., "On Removing Routing Protocol from Future Wireless Networks: A Real-Time Deep Learning Approach for Intelligent Traffic Control," EEE Wireless Communications, 2017. Available: https://ieeexplore.ieee.org/document/8088549

[12] Nicholas B. LaFarge, et al., "Autonomous closed-loop guidance using reinforcement learning in a low-thrust, multi-body dynamical environment," Acta Astronautica, 2021. Available: https://www.sciencedirect.com/science/article/abs/pii/S0094576521002460

[13] Yingnong Dang, et al., "IOps: Real-World Challenges and Research Innovations," 2019

IEEE/ACM 41st International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), 2019. Available: https://ieeexplore.ieee.org/document/8802836

[14] Ahmad Asghar, et al., "Self-Healing in Emerging Cellular Networks: Review, Challenges, and Research Directions," IEEE Communications Surveys & Tutorials, 2018. Available: https://ieeexplore.ieee.org/abstract/document/8335292

[15] Aris Leivadeas and Matthias Falkner, "A Survey on Intent-Based Networking," IEEE Communications Surveys & Tutorials, 2023. Available: https://ieeexplore.ieee.org/document/9925251

[16] Robert R. Hoffman, et al., "Metrics for Explainable AI: Challenges and Prospects," arXiv, 2018. Available: https://arxiv.org/pdf/1812.04608

[17] Philip Koopman and Michael Wagner, "Autonomous Vehicle Safety: An Interdisciplinary Challenge," IEEE Intelligent Transportation Systems Magazine, 2017. Available: https://ieeexplore.ieee.org/document/7823109

[18] Eric J. Topol, "High-performance medicine: the convergence of human and artificial intelligence," Nature Medicine, 2019. Available: https://www.nature.com/articles/s41591-018-0300-7

[19] Stefano Zanero, "When Cyber Got Real: Challenges in Securing Cyber-Physical Systems," 2018 IEEE SENSORS, 2018. Available: https://ieeexplore.ieee.org/document/8589798

[20] Tian Li, et al., "Federated Learning: Challenges, Methods, and Future Directions," IEEE Signal Processing Magazine, 2020. Available: https://ieeexplore.ieee.org/document/9084352

[21] Aws Albarghouthi, "Introduction to Neural Network Verification," arXiv, 2021. Available: https://arxiv.org/pdf/2109.10317

[22] Ching-Nam Hang, et al., "Large Language Models Meet Next-Generation Networking Technologies: A Review," Future Internet, 2024. Available: https://www.mdpi.com/1999-5903/16/10/365

[23] Kilian Q. Weinberger, et al., "Distance Metric Learning for Large Margin Nearest Neighbor Classification," Journal of Machine Learning Research, 2009. Available: https://jmlr.csail.mit.edu/papers/volume10/weinberger09a/weinberger09a.pdf

[24] Scott Rose, et al., "Zero Trust Architecture," NIST Special Publication 800-207, 2020. Available: https://nvlpubs.nist.gov/nistpubs/specialpublications/NIST.SP.800-207.pdf

[25] Jeffrey O. Kephart and David M. Chess, "The Vision of Autonomic Computing," IEEE Computer, 2003. Available: https://ieeexplore.ieee.org/document/1160055