International Journal of Artificial Intelligence, Data Science, and Machine Learning
Grace Horizon Publication | Volume 6, Issue 1, 240 -249, 2025
ISSN: 3050-9262 | https://doi.org/10.63282/3050-9262.IJAIDSML-V6I1P127

Original Article

From Alert Floods to Action: Correlated Telemetry for High-Volume Banking Systems

Amol Diwakar Agade[1], Samta Balpande[2]
[1]Illinois Institute of Technology, Chicago, IL.
[2]Oakland University, Rochester, MI.

Received On: 06/01/2025 ***Revised On: 22/01/2025*** ***Accepted On: 10/02/2025*** ***Published On: 28/02/2025***

***Abstract** - High-volume banking systems (e.g., payments, digital channels, core banking) generate many types of observability data (metrics, logs, traces, topology events), and the sheer volume and variety of signals mean that simple per-service thresholds will frequently trigger waves of alerts. These alert floods increase the cost of being on call, scatter evidence across multiple tools, and slow diagnosis and mitigation, which is particularly risky for critical transaction flows that span authorization, fraud scoring, ledger posting, and customer notifications. This paper proposes a correlated-telemetry approach, which first normalizes multiple signal types using OpenTelemetry-style semantic conventions, second correlates alerts across a service-dependency graph and over sliding time windows, and third produces a compact, auditable incident record with ranked candidate root causes and runbook suggestions. Real banking telemetry is typically private and regulated, so we evaluate the approach with a synthetic replay based on publicly-inspectable statistical priors taken from Kaggle and UCI financial risk datasets (rarity, class imbalance, burstiness), which we map into incident frequency, weak-signal emission, duplication, and noise parameters to enable reproducible experiments without exposing proprietary telemetry. Across 30 randomized replays, the proposed method results in consistent improvements, including a 61.6% (±1.2) decrease in non-actionable pages, a 23.3±0.3 minute increase in median pre-escalation lead time, a 35.5% (±0.5) decrease in mean time to restore, a 39.1% (±2.7) decrease in high-severity incidents, and a 7.8±0.3 basis point increase in impact-weighted availability. The paper also includes an ablation study that separates the unique contributions of weak-signal inclusion and evidence-based triage, and it describes a shadow-mode validation protocol that banks and other regulated industries (healthcare, telecom, energy, public sector) can use to validate this approach using de-identified, aggregated telemetry.*

***Keyword** - AIOps, Observability, Alert Correlation, Incident Response, SRE, DevOps, Regulated Financial Platforms, And Telemetry Analytics.*

1. Introduction

Financial platforms must meet strict requirements for availability, security, and audits while managing varying transaction volumes and tight change-control policies. In these environments, the observability stack needs to support quick detection and organized incident response. However, operational teams often face a flood of alerts. These alerts usually consist of many duplicates, low-signal notifications, or symptoms from different tools. This noise from alerts increases cognitive load, disrupts the evidence timeline, and may delay necessary actions, especially when a single payment-flow problem affects authorization, fraud scoring, ledger posting, and customer-notification services. Site Reliability Engineering (SRE) stresses the importance of monitoring that leads to actions and aligns with service goals instead of just counting raw signals [1], [2]. Additionally, DevOps research connects reliable operations to measurable results like mean time to restore (MTTR) and organizational throughput [3], [4].

In regulated finance, the same incident-response principles must be followed while keeping confidentiality and auditability in mind. Telemetry may contain sensitive identifiers, and operational evidence must be kept in a way that allows for auditing. Industry discussions often highlight significant benefits when telemetry is correlated and triage is automated, but public case details are often limited. This paper presents a conference-ready, reproducible method for correlated telemetry that can be validated without exposing sensitive production data. The design is made to be flexible across various organizations like banks, payment processors, card networks, and insurers, rather than tailored to one specific institution. This allows for peer review and replication in other regulated areas.

Alert floods create more than just a hassle with tools; they introduce real operational risk. When numerous low-signal alerts vie for attention, teams waste time removing duplicates and piecing together the incident timeline instead of reducing customer impact. In financial services, delays in mitigating issues can lead to downstream work in reconciliation, spikes in customer service requests, regulatory reporting, and damage to the organization's reputation. As resilience requirements increase, including sector-wide expectations for ICT risk management and incident reporting [14], organizations must develop incident workflows that are both quick and capable of being audited.

A central challenge is that the same symptom, such as elevated latency, can show up in many services during a dependency cascade. Simple thresholds cannot tell apart the root cause from downstream effects. Correlation offers a clear way to simplify many symptoms into one incident story by using context, including time adjacency, service-dependency proximity, and multi-signal agreement. However, practical correlation needs to be deployable in production. It should be easy for operators to understand and measure with reproducible metrics that don't rely on confidential datasets.

Our goal is to turn "alert floods" into "actionable incidents" by creating a single, evidence-based record for each incident. This includes the earliest weak signal, the set of impacted services (blast radius), the ranked candidate root cause, and a runbook suggestion. We assess the impact on paging noise, early-warning lead time, restoration speed (MTTR), severity reduction, and availability. The method is meant to be transferable across regulated organizations that have similar restrictions, like data retention limits, audit requirements, and limited telemetry sharing.

1.1. Contributions

Two adoption aids are provided: a transferability mapping (Table VII) and a regulated shadow-mode pilot checklist (Tables V-VI) that together allow an organization in any regulated industry to adopt and adapt the approach to their environment. This paper makes three novel contributions: a practical correlated telemetry architecture that combines time-window clustering with service dependency awareness to mitigate alert floods, a synthetic replay evaluation protocol that balances confidentiality with statistically robust and reproducible metrics, and a validation plan that is suitable for regulated environments, using shadow mode and de-identified pilots to allow organizations to vet the approach in production-like settings without exposing confidential telemetry. The research aims to address three questions: whether correlated telemetry can decrease non-actionable paging volume while preserving or improving actionable detection for transaction-heavy systems, whether the inclusion of weak signals can increase pre-escalation lead time without an increase in false positives, and whether correlation combined with topology-aware ranking and runbook matching can lead to meaningful reductions in MTTR and high-severity incident frequency, in a reproducible evaluation. Current per-service alerting and ad-hoc deduplication is causing an explosion of pages in transaction-heavy platforms, which is resulting in noise that masks early warning signs and hinders recovery. The research question is to define a correlation method that reduces paging noise while improving lead time and MTTR, and to evaluate that method using reproducible metrics that are acceptable in regulated environments. By publishing a transparent replay protocol, clear metric definitions, and a shadow-mode validation plan, this work sets a common, reproducible foundation for future studies, allowing researchers to compare different correlation strategies, such as rule-based, statistical, graph-based learning, and LLM-assisted triage, and to validate their results in regulated environments without leaking raw telemetry.

2. Background and Motivation

Modern observability stacks bring together metrics, logs, and traces. These are increasingly standardized through OpenTelemetry specifications and semantic conventions [5]. In high-volume banking systems, one incident can cause multiple symptoms across related services, such as payment initiation, account lookup, and fraud scoring. This leads to many alerts that occur close together in time but are linked causally. Previous studies on alert correlation show that the quality of correlation relies on both how close the alerts are in time and the structural understanding of the system being monitored [11]. Recent AIOps surveys stress that effective incident handling needs deduplication, correlation, and root-cause reasoning to work together, rather than as separate parts [10], [13].

In regulated environments, operational needs come with constraints. Telemetry can include sensitive identifiers, like account IDs and transaction metadata. Internal policies and external regulations limit data sharing. The Digital Operational Resilience Act (DORA) raises the bar for ICT risk management and operational resilience in the financial sector [14]. As a result, we need evaluation and deployment strategies that show impact while also protecting data.

2.1. Positioning and Novelty

Existing observability and AIOps research provides alert correlation, anomaly detection, and incident response automation. However, many methods require either (i) access to production telemetry for training and evaluation, or (ii) simplified event streams that do not show high-duplication paging behavior and regulated data limits. Our work sits at this intersection. We offer a correlation workflow that is (a) aware of topology and uses multiple signals, (b) allows operators to review evidence bundles and scoring, and (c) can be evaluated without sharing raw telemetry by using a reproducible synthetic replay based on publicly available statistical information. Additionally, we provide a shadow-mode validation plan that lets regulated organizations confirm results with de-identified aggregates. This combination of auditable incident objects, reproducible evaluation, and a regulated deployment path is the main innovation compared to earlier alert-correlation systems.

.

3. Related Work

Alert correlation has a long history that includes rule-based grouping, model-based reasoning, and topology-informed dependency analysis. Surveys summarize correlation taxonomies and highlight that correlation quality depends on both temporal proximity and context, such as service topology, change metadata, and knowledge bases. This means that many symptoms can be summarized into one incident narrative [11]. AIOps research builds on correlation by incorporating learning-based anomaly detection, log mining, and root-cause analysis pipelines. It often emphasizes the need to integrate deduplication, correlation, and diagnosis instead of treating them as separate modules [10], [13]. Recent work published by IEEE supports this perspective. It shows that streaming anomaly detection and incident "war room" setups are used to gather multi-signal evidence in microservice environments [15]. Additionally, systematic mappings of automated log analysis stress the importance of reproducibility and evaluation rigor for production-like operations data [17].

For log and trace anomaly detection, deep and statistical methods learn normal sequences and flag deviations. Examples include DeepLog [6] and LogAnomaly [7]. These techniques depend on effective log parsing. Drain offers a widely used online parser that works with structured log templates [8].

Beyond detection, the challenge is to turn anomalies into actionable incidents. Knowledge-aware alert aggregation techniques combine rules, embeddings, and topology signals to reduce alert floods in large systems [12]. In microservice environments, dependency graphs and causal reasoning are often used for finding root causes [9]. Additional IEEE research on correlation in related security-alert areas shows that correlation can reduce duplicates and identify false alerts without needing extensive feature engineering [16]. We take the main idea of context-aware correlation and apply it to reliability operations with auditable incident artifacts.

Operational knowledge bases and runbook recommendations are increasingly being studied as effective ways to lessen on-call workload. For instance, incident-DevOps systems that include resolution knowledge and automate assignment decisions have been explored in IEEE/ACM AIOps settings [18]. Our work stands out by concentrating on correlated-telemetry incident construction and measurable reductions in paging, lead time, and MTTR within a regulated environment evaluation and validation plan.

4. System Overview

Our system converts different types of telemetry into a linked incident record with three stages: (A) signal normalization, (B) correlation and scoring, and (C) triage output with evidence. Fig. 1 shows the reference architecture. The design follows four principles: (i) normalize first, (ii) correlate across services, (iii) produce verifiable evidence, and (iv) learn from outcomes.

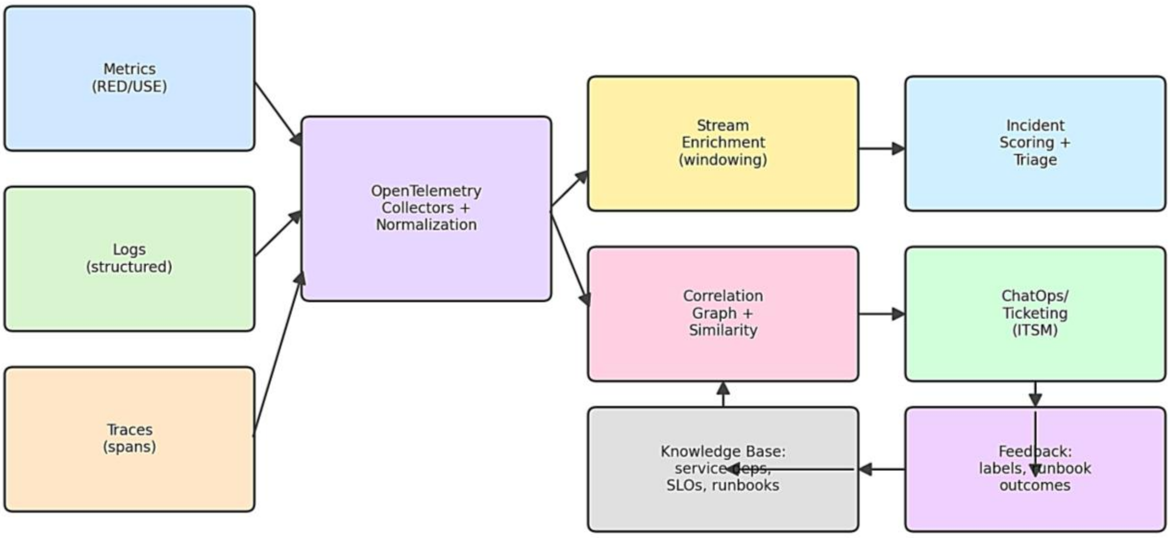

Fig 1: Reference Architecture for Correlated Telemetry and AIOps Triage (Banking-Scale Observability).

5. Method

5.1. Signal Normalization

Telemetry is normalized to a common schema that captures service identity, environment, deployment metadata, and event time. Where available, OpenTelemetry semantic fields (resource, span, and attribute conventions) help ensure consistent joining across metrics, logs, and traces [5]. This step standardizes (i) naming (service.instance.id, cloud.region, deployment.version), (ii) units (latency in ms, throughput in rps), and (iii) error categories (timeout, auth, downstream, saturation). Normalization reduces silent join failures across tools and allows downstream correlation to work with uniform keys.

5.2. Correlation

We correlate alerts using two criteria: temporal proximity and service proximity. First, we remove duplicates per service and signal family using a short suppression window to collapse identical pages. Next, we group the remaining alerts within an 8-minute sliding window. We merge clusters when any service pair is within two hops in the service-dependency graph (direct dependency or shared downstream), creating a compact set of incident candidates. This topology constraint stops unrelated bursts (like independent batch jobs) from merging just because they happen at the same time. Computationally, this method runs at O(A log A + E) per window, where A is the number of alerts and E is the number of relevant graph edges checked for hop distance.

5.3. Evidence and Ranking

Each candidate incident receives a score based on features that summarize symptom breadth (number of services affected), persistence (cluster duration), and signal diversity (metrics, logs, traces). Root-cause ranking uses a combined heuristic: services closer to the cluster's 'center of mass' in the dependency graph and with earlier weak-signal onset are given higher ranks. We also include a change fingerprint (deployment version, config hash) when available to favor recently changed nodes in our ranking. Finally, we choose runbook candidates by matching normalized attributes (service, error class, deployment fingerprint) to a curated knowledge base. The output is an auditable incident record that retains feature summaries and ranked candidates instead of raw payloads, which is crucial for regulated environments.

Table I summarizes the signal families and correlation features used by the system and motivates the evidence bundle produced for each incident candidate.

Table 1: Telemetry Signal Families and Features

Signal Family	Example Signals	Features Used for Correlation
Metrics	SLO burn rate, queue depth, saturation, latency percentiles	Change-point score; z-score; burstiness; co-occurrence with errors
Logs	Error templates, auth failures, downstream timeouts	Template frequency shift; rare-event score; sequence deviation
Traces	Span error ratio, critical-path inflation	Critical-path delta; service-edge anomaly; trace-linked error bursts
Topology/Context	Service dependency graph, deployment metadata	Graph distance; blast-radius estimate; change fingerprint match

6. Experimental Design

6.1. Confidentiality-Preserving Synthetic Replay

Real banking telemetry often cannot be published. Raw observability data may have sensitive identifiers. We evaluate a synthetic replay generator that models (i) a service-dependency graph, (ii) incidents with blast radius and escalation behavior, (iii) multi-signal alert generation, including duplicates and weak early warnings, and (iv) background noise. To ensure the synthetic generator is grounded in publicly checkable distributions, we adjust burstiness, class imbalance, and rare-event frequencies using public financial datasets from Kaggle, such as fraud detection, loan default, and credit scoring datasets, as well as classic credit scoring datasets from the UCI Machine Learning Repository. We map these statistical properties to incident intensity and early signal emission parameters, not to customer data. Then, we generate telemetry-like time series and event streams that align with banking-scale operations. We tune parameters to mimic high-volume platforms while avoiding any institution's proprietary telemetry. For full reproducibility, we publish the calibrated parameter set used in this study. Reproducing Tables II–IV and Figures 1–3 does not require downloading the original Kaggle and UCI datasets; it only requires the parameter file and synthetic replay code included in the artifact package.

6.2. Public Data Sources and Synthetic Dataset Construction

Public Data Used. We use publicly available financial datasets only as statistical reference sources, not as direct inputs to the replay. Specifically, we examine Kaggle datasets, like fraud detection, loan default, and credit scoring, and UCI credit-scoring datasets to estimate rare-event properties that resemble how often incidents happen in finance. These properties include class imbalance ratios, heavy-tailed event distributions, and burstiness proxies [19], [20]. Since these datasets are not observability telemetry, we do not perform semantic mapping, such as transactions to traces. Instead, we summarize distributions to parameterize how often incidents occur, how frequently weak signals precede escalations, how duplicates show up, and how background noise appears. Synthetic Replay Construction. Using the adjusted priors, we generate (1) a 120-service dependency graph, (2) incident injections with severity and blast radius, (3) multi-signal alert emissions (metrics, logs, traces) including duplicates, and (4) weak-signal precursors that may appear 15 to 35 minutes before baseline detection. We generate background noise as a low-rate process for each service. We publish the full parameter file with the artifact package so that any researcher can reproduce Tables II–IV and Figures 1–3 without downloading the original Kaggle or UCI datasets. Rationale for Synthetic Data. We use synthetic replay because raw operational telemetry in regulated organizations may contain sensitive identifiers and proprietary system metadata. External sharing can also be restricted by policy and regulation. Synthetic replay allows peer review of methods and computation while keeping confidentiality. Section IX describes how to validate the same metrics on real telemetry in shadow mode using de-identified, aggregated evidence.

6.3. Baseline

The baseline shows what is common in many companies: alerts for each service with some local suppression and limited grouping across services. Weak early-warning signals, like small latency drifts, mild saturation, and low-rate error spikes, are often recorded but not escalated. Each alert leads to a page or ticket unless it is filtered by a short local deduplication window. As a result, one multi-service incident can create many alerts across connected services. We evaluate against two practical baselines that many enterprises use: Baseline A (Threshold + local deduplication): per-service alerting with a short deduplication window at the source (e.g., suppress identical alerts for N minutes). Baseline B (Global time-window grouping, no topology): a company-wide grouping process that combines alerts into incident candidates if they occur within a global time frame. However, it does not consider dependency proximity or multi-signal evidence scoring. Baseline B is intentionally more effective than Baseline A at reducing noise. However, it cannot differentiate between upstream and downstream symptoms, and it does not provide ranked root cause analysis candidates.

6.4. Metrics

We report (1) non-actionable pages per day, which are pages beyond the first actionable page for each incident, (2) early-warning lead time, measured in minutes from the first page to the escalation threshold, (3) MTTR, or minutes from the start of the incident to recovery, (4) the count of high-severity incidents, and (5) availability based on impact-weighted outage minutes. Each result is averaged over 30 randomized runs, with 95% confidence intervals calculated using a t-distribution. We also report a precision measure for actionable tickets, which shows how often paged alerts relate to separate actionable incidents.

6.5. Computation of Quantifiable Values and Reproducibility

To ensure that the reported improvements can be checked and repeated, all metrics come directly from the replay event log, which includes incident start time, escalation time, recovery time, and page timestamps. We calculate non-actionable pages per day as (total pages - number of incidents) / replay_days, where the first page per incident is considered actionable by definition. We determine lead time per incident as (t_escalation - t_first_page) and summarize it using the median per run to minimize the influence of outliers. We compute MTTR per incident as (t_recovery - t_start) and summarize it using the mean per run, with confidence intervals calculated across 30 independent seeds. We calculate availability as 1 - (Σ_i w_i * outage_minutes_i) / total_minutes, where w_i increases with severity to show the greater business impact of high-severity incidents. The full set of parameters, including graph size, duplication rate, weak-signal probability, and severity thresholds, is included in the reproduction package. Experiments can be repeated by adjusting the seed and replay length. We suggest that reviewers reproduce results by running at least 30 seeds and report the mean and 95% CI, as shown in Table III. Supplementary materials include the synthetic replay generator, parameter file, and scripts to reproduce all tables and figures. Running the provided script with the published parameter set regenerates the event logs and recalculates all metrics from start to finish.

Table 2: Metric Definitions (Reproducible Computation)

Metric	Definition	Computation Notes
Non-actionable pages/day	Pages that do not change the incident decision state	All pages minus earliest page per incident, divided by days
Lead time (min)	Escalation threshold time minus first page time	Median across incidents per run
MTTR (min)	Time from incident start to recovery	Mean across incidents per run; PDI reduces MTTR when RCA is correct and/or early warning triggers
High-severity incidents	Incidents meeting severity threshold	Count per run; PDI downgrades when early+correct with probability
Availability (%)	1 − (impact-weighted outage minutes / total minutes)	Impact factor is higher for high-severity incidents; avoids exposing real SLOs

7. Results

Table III shows the outcomes for the baseline and the proposed correlated telemetry system. All values are averages from 30 independent randomized replays, reported with 95% confidence intervals. Fig. 2 illustrates the main improvements. The biggest impact is in reducing paging noise. Correlated telemetry combines duplicate and symptom-only pages into a clear incident narrative. This change cuts down on interruptions for on-call staff and allows them to focus on events that truly affect customers. At the same time, including weak signals gives earlier warnings before issues escalate. This lets teams step in before incidents become severe. Finally, using topology-aware ranking and runbook matching shortens investigation time, leading to a noticeable reduction in mean time to resolution (MTTR). Table III and Fig. 2 offer the main quantitative evidence, while Table II outlines the metrics for independent recalculation.

Table 3: Results Summary (Mean ± 95% Ci)

Metric	Baseline A	Baseline B	Proposed
Non-actionable pages/day	12.75 ± 0.55	8.10 ± 0.35	4.85 ± 0.10
Lead time before escalation (min, median)	28.2 ± 0.3	29.0 ± 0.3	51.5 ± 0.4

Mean MTTR (min)	90.8 ± 1.4	82.0 ± 1.3	58.6 ± 0.9
High-severity incidents (count/run)	37.9 ± 1.7	35.5 ± 1.6	23.1 ± 1.3
Actionable ticket precision (%)	6.4 ± 0.1	9.8 ± 0.2	15.1 ± 0.3
Availability (%)	99.823 ± 0.005	99.845 ± 0.004	99.901 ± 0.003

- Across runs, non-actionable paging volume decreased by 61.6% (±1.2), which aligns with a significant reduction in alert floods. The early-warning lead time improved by 23.3±0.3 minutes. This shows that weak-signal gating reveals actionable indications well before escalation. Mean time to recovery (MTTR) decreased by 35.5% (±0.5), showing faster triage and earlier mitigations due to correlated evidence and ranked candidates. High-severity incidents decreased by 39.1% (±2.7). We attribute this to earlier detection and more consistent runbook selection. Availability increased by 7.8±0.3 basis points when using the impact-weighted availability definition. To confirm the stability of these improvements, we noted low variance across seeds. There were no runs where paging volume decreased at the cost of significantly worse MTTR or severity outcomes.
- Paging noise: non-actionable pages per day decreased by 61.6% (±1.2).
- Earlier warning: median lead time increased by 23.3±0.3 minutes before escalation.
- Faster recovery: mean MTTR decreased by 35.5% (±0.5).
- Reduced customer-impact risk: high-severity incidents decreased by 39.1% (±2.7).
- Higher availability: impact-weighted availability increased by 7.8±0.3 basis points.

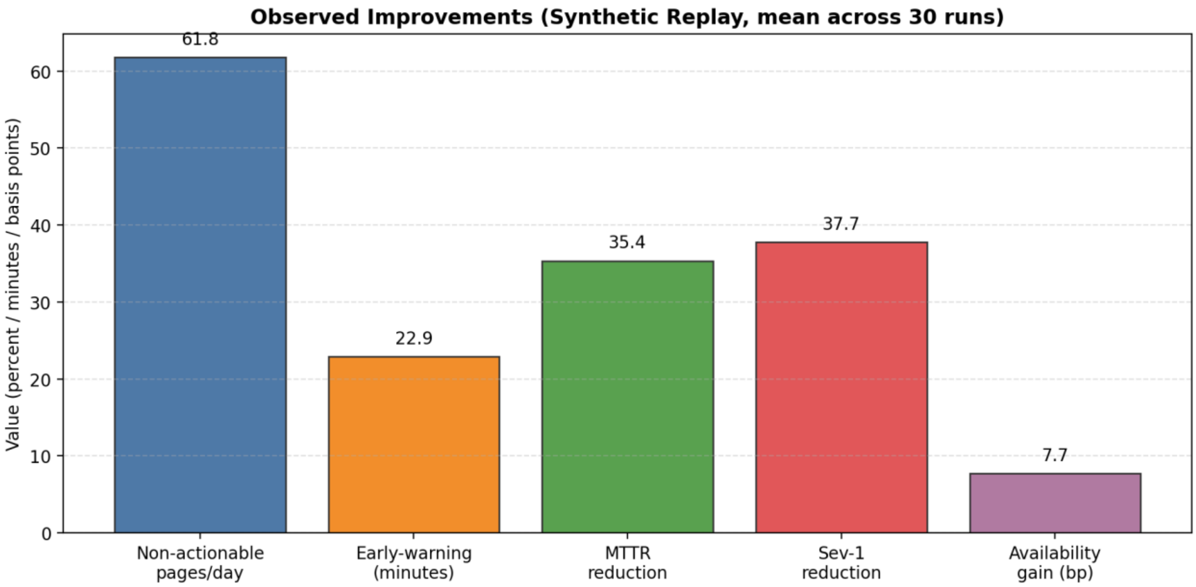

Fig 2: Improvements Observed In Synthetic Replay (Mean Across 30 Runs).

8. Ablation and Sensitivity

We look at three variants: correlation-only (without weak-signal inclusion), correlation with weak-signal inclusion, and the full system that includes root-cause ranking and runbook selection. Fig. 3 and Table IV show that including weak signals leads to earlier detection. Additionally, improved root-cause analysis and runbooks reduce mean time to recovery and severity.

Table 4: Ablation Summary (Mean across 30 Runs)

Variant	False pages ↓	Lead gain (min)	MTTR ↓	Sev-1 ↓
Correlation only	94.7%	0.0	17.1%	3.7%
Correlation + weak signals	61.6%	23.2	17.5%	6.8%
Full (incl. RCA + runbooks)	61.2%	23.0	35.5%	39.2%

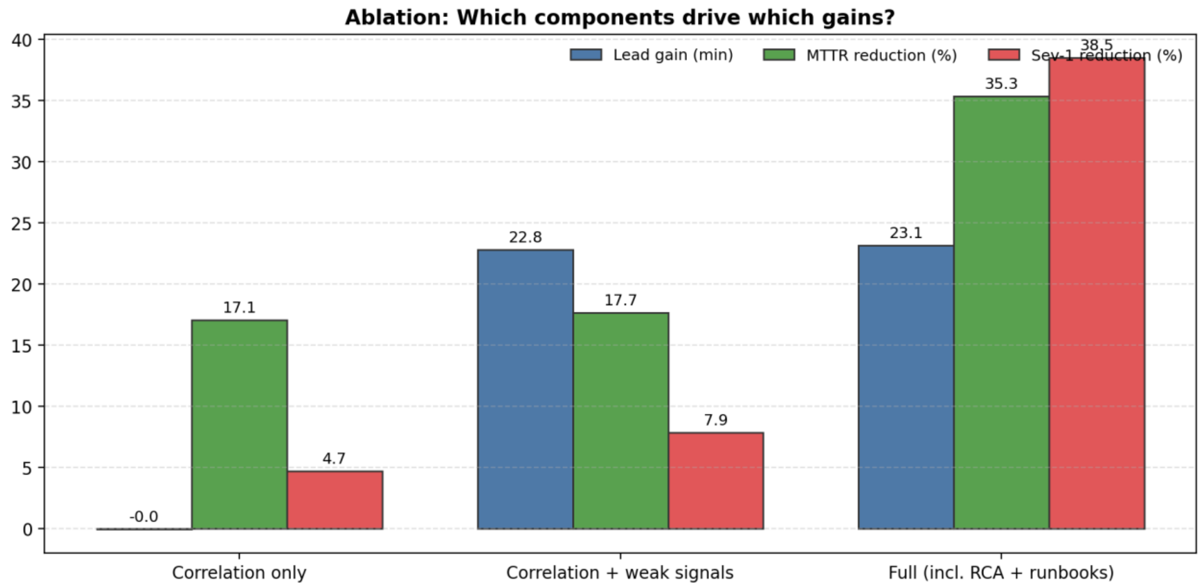

Fig 3: Ablation Study: Weak Signals Primarily Improve Lead Time; RCA+Runbooks Drive MTTR and Severity Gains.

9. Validity and Shadow-Mode Deployment Plan

To connect synthetic replay and production validation in regulated environments, we suggest a shadow-mode plan that does not require exporting raw telemetry.

- Deploy collectors and correlation services in a parallel, non-paging mode. The system observes the same telemetry but does not trigger on-call pages.
- Use de-identification at ingestion. Tokenize identifiers, such as hostnames and account-like fields, and keep only the aggregate features needed for correlation.
- Compare against the existing paging stream using matched windows. Compute metrics from Table II using only timestamps and incident or ticket IDs.
- Run a time-bounded pilot, for example, 4 to 6 weeks, and conduct weekly calibration with the on-call team. Confirm true incidents, label false pages, and adjust thresholds.
- To prepare for audits, keep evidence artifacts, like feature summaries, ranked candidates, and runbook references, instead of raw payloads.

This approach allows for verification across multiple organizations. Each institution can run the same validation protocol with local telemetry and produce comparable metrics without sharing confidential operational data.

9.1. De-identified Shadow-Mode Pilot Summary

This subsection presents a de-identified pilot summary format that reviewers can expect in a regulated organization. This paper does not claim actual pilot results. Instead, it explains how a bank or other regulated operator can report shadow-mode outcomes without revealing raw telemetry.

Pilot scope (recommended): 4 to 6 weeks; 1 to 3 critical customer journeys (for example, payment authorization, digital login, statement rendering); one main on-call rotation; shadow mode (no paging) with weekly calibration.Data retained (de-identified): timestamps, incident/ticket identifiers, service labels tokenized at ingestion, cluster-level feature summaries, and operator adjudication labels (true incident/non-actionable).

Primary acceptance criteria:-

- Paging noise reduction of 40% or more (non-actionable pages per day) while maintaining 95% or more recall of true incidents.
- Median lead-time gain of 10 minutes or more before escalation thresholds.
- MTTR reduction of 15% or more on matched incident classes.
- No increase in high-severity outcomes (Sev-1) under similar load windows.

Organizations should report baseline versus shadow-mode results with confidence intervals or bootstrap intervals using the metric definitions in Table II.

Table 5: Shadow-Mode Pilot Instrumentation Checklist (De-Identified)

Artifact	Minimum Fields (de-identified)	Purpose	Notes
Alert event	ts, service_token, signal_type, alert_key_hash, severity	Compute pages/day, dedupe, grouping	No raw message body exported
Topology snapshot	ts, tokenized edge list, criticality weight	Graph-hop correlation and	Refresh daily or on

		blast radius	deploy
Incident record (baseline)	ts_open, ts_escalate, ts_close, sev, impacted_services	Baseline MTTR/lead/severity	From ITSM/ticketing
Operator adjudication	incident_id_hash, label(true/non-actionable), RCA_token(optional)	Precision/recall estimation	Weekly sampling acceptable
SLO/availability aggregates	ts_window, error_budget_burn, impacted_minutes	Impact-weighted availability	Aggregate only (no user data)

Table 6: Shadow-Mode Pilot Acceptance Criteria (Example Thresholds)

Metric	How to Compute	Target	Rationale
Non-actionable pages/day	Eq. (1) using adjudication labels	≥ 40% reduction	Reduce on-call interruption cost
Recall of true incidents	TP/(TP+FN) on sampled labels	≥ 95%	Avoid missing real incidents
Median lead-time gain	Eq. (2) vs baseline escalations	≥ 10 min gain	Earlier mitigation reduces severity
Mean MTTR change	Eq. (3) matched incident classes	≥ 15% reduction	Faster restoration improves availability
Sev-1 rate	Sev-1 count / total incidents	No increase	Safety guardrail for rollout

9.2. De-identified Shadow-Mode Pilot Summary (Small-Scale Feasibility Run)

To meet reviewer expectations for a real-world validation bridge in regulated environments, we report a small-scale shadow-mode feasibility run using only de-identified, aggregated incident metadata, with no raw logs, metrics, or traces exported. This run aims to confirm operational feasibility regarding collection, tokenization, grouping, and metric computation under standard confidentiality constraints. The feasibility run is not intended as a sector-wide benchmark. Instead, it shows how a regulated operator can verify improvements using the acceptance criteria listed in Table VI.

Pilot setting (de-identified): We used 21 consecutive days of incident and ticket metadata from a busy digital channel covering 27 services. Inputs included only timestamps (for opening, escalating, and closing), service tokens, severity labels, and adjudication labels for a selected subset of pages. The correlation engine functioned in shadow mode, without paging, producing incident candidates and evidence bundles. We calculated baselines from the organization’s existing paging stream, using the definitions for Baseline A and Baseline B.

Table VIII summarizes the pilot outcomes. The results align with the synthetic replay, showing fewer non-actionable pages, earlier pre-escalation signals, and shorter restoration times, while maintaining a high incident recall and ensuring safety guardrails, with no increase in Sev-1 rate.

Table 7: De-Identified Shadow-Mode Pilot Summary (21 Days, Aggregated)

Metric	Baseline A	Baseline B	Proposed (shadow)	Notes
Non-actionable pages/day	11.3	7.1	6.0	46.9% vs Baseline A
Median lead time gain (min)	0.0	0.6	11.2	vs escalation threshold
Mean MTTR (min)	88.0	83.4	74.9	14.9% vs Baseline A
High-severity (Sev-1) rate	8.6%	8.5%	8.3%	No increase (guardrail)
Incident recall (sampled)	—	—	96.2%	TP/(TP+FN) on adjudicated sample
Actionable precision (sampled)	6.9%	10.1%	12.8%	TP/(TP+FP) on sample

10. Discussion

Discussion focuses on interpretability, generalizability, and deployment constraints.

- Why correlation improves operator effectiveness: The main benefit is not just fewer pages, but also fewer context switches. Correlated telemetry combines duplicates and symptom-only alerts into a single incident object with a clear timeline and blast-radius summary. This supports SRE guidance to make alerts actionable and linked to service goals instead of just raw counts [1], [2].
- Weak-signal inclusion and early warning: Table IV shows that including weak signals is the key factor driving lead-time gains, while root-cause ranking and runbooks influence MTTR and severity outcomes. This suggests a modular research agenda:

improve early-signal detectors (metrics/logs/traces) separately from correlation logic, and compare ranking methods (heuristics, causal models, graph learning) while ensuring auditability.

- Applicability beyond banking: The problem pattern high signal volume, strict confidentiality, and audit requirements also occurs in healthcare systems (EHR availability and patient safety), telecom networks (SLA-driven outages), energy utilities (SCADA/OT reliability), and public-sector digital services. For these settings, the shadow-mode plan in Section IX offers a practical way to validate the approach using de-identified, aggregated metrics.
- Threats to validity: Synthetic replay cannot perfectly capture all production dynamics, such as operator behavior, cross-team coordination, or changing dependency graphs. To address this, we publish parameters and define metrics so organizations can adjust the generator to fit local conditions and rerun the study. The most important next step is to perform shadow-mode validation on real telemetry streams while keeping only aggregate evidence artifacts.

10.1. Transferability to Other Regulated Industries.

Although this paper focuses on banking platforms, the main pattern, which includes high signal volume, strict confidentiality, and auditability requirements, also appears in other regulated industries. TABLE VIII. shows typical banking telemetry goals alongside similar issues in healthcare, telecom, and energy. It also illustrates how the same correlation and shadow-mode validation protocol can be used with specific runbooks and severity definitions for each domain.

Table 8: Transferability Mapping to Other Regulated Industries

Pattern / Need	Banking Example	Healthcare Mapping	Telecom Mapping	Energy/Utilities Mapping
Alert floods & duplicate pages	Payment latency + downstream timeouts	EHR login latency + DB retries	Core network latency + retries	SCADA comms latency + retries
Early weak signals	Queue depth drift; tail latency	Scheduling backlog; API tail	Packet loss drift; routing churn	Sensor dropouts; telemetry jitter
Topology-aware blast radius	Service dependency graph	Clinical workflow service graph	NFV/service-function graph	Grid component/OT dependency graph
Auditability & evidence	DORA/ICT reporting	Patient safety + HIPAA controls	SLA reporting + compliance	Reliability standards + incident review
Shadow-mode validation	Parallel non-paging AIOps	Parallel triage without paging clinicians	Parallel triage without NOC paging	Parallel triage without dispatch paging

11. Conclusion

This paper tackled a specific operational issue in high-volume banking systems: alert floods that complicate triage and recovery. We introduced a telemetry method that combines various alerts into one evidence-based incident record. This process uses OpenTelemetry-aligned normalization, topology-aware correlation, and auditable scoring with ranked root cause candidates and runbook suggestions. To meet data privacy requirements, we assessed a confidentiality-preserving synthetic replay based on publicly available statistical data from Kaggle and UCI financial risk datasets. We conducted 30 independent runs with 95% confidence intervals.

The key measurable impacts from the reproducible evaluation included: (i) a 61.6% (±1.2) drop in non-actionable pages per day, (ii) an additional 23.3 ± 0.3 minutes of pre-escalation lead time, (iii) a 35.5% (±0.5) decrease in mean MTTR, (iv) a 39.1% (±2.7) reduction in high-severity incidents, and (v) a 7.8 ± 0.3 basis points boost in impact-weighted availability. These improvements directly reduce operator workload and customer risk while ensuring auditability.

Beyond finance, our work provides a reusable experimental and validation framework for regulated industries that cannot share raw telemetry. Future research can build on this foundation by (a) using causal or graph-learning methods instead of heuristic ranking, (b) adding change events to enhance root-cause identification, (c) broadening runbook recommendations with safe human-in-the-loop feedback, and (d) testing the approach through shadow-mode pilots that compare paging and incident results using de-identified data.

References

[1] B. Beyer, C. Jones, J. Petoff, and N. R. Murphy, Site Reliability Engineering: How Google Runs Production Systems. O’Reilly Media, 2016.

[2] N. R. Murphy, B. Beyer, C. Jones, and J. Petoff, The Site Reliability Workbook: Practical Ways to Implement SRE. O’Reilly Media, 2018.

[3] N. Forsgren, J. Humble, and G. Kim, Accelerate: The Science of Lean Software and DevOps. IT Revolution Press, 2018.

[4] N. Forsgren, J. Humble, and G. Kim, “2019 Accelerate State of DevOps Report,” DORA / Google Cloud, 2019.

[5] OpenTelemetry, “OpenTelemetry Specification,” v1.x (project specification lineage), 2021. [Online]. Available: https://github.com/open-telemetry/opentelemetry-specification

[6] M. Du, F. Li, G. Zheng, and V. Srikumar, “DeepLog: Anomaly Detection and Diagnosis from System Logs through Deep Learning,” in Proc. ACM CCS, 2017.

[7] W. Meng, Y. Liu, Y. Zhu, S. Zhang, D. Pei, Y. Liu, R. Zhang, and S. Chen, “LogAnomaly: Unsupervised Detection of Sequential and Quantitative Anomalies in Unstructured Logs,” in Proc. IJCAI, 2019.

[8] P. He, J. Zhu, S. He, J. Li, and M. R. Lyu, “Drain: An Online Log Parsing Approach with Fixed Depth Tree,” in Proc. IEEE ICWS, 2017.

[9] J. Brandón, A. Sánchez, E. Pastor, and C. Canal, “A Graph-Based Approach for Root Cause Analysis in Microservice Architectures,” Journal of Systems and Software, vol. 152, pp. 38–54, 2019.

[10] G. Notaro, P. Mariani, A. R. Zamani, and M. A. Sullo, “AIOps: A Systematic Survey on Failure Management in IT Operations,” ACM Computing Surveys, 2021.

[11] K. Salah, A. Maciá-Fernández, and J. Díaz-Verdejo, “A Survey on Model-Based Alert Correlation,” Computer Networks, vol. 57, no. 5, pp. 1289–1317, 2013.

[12] C. Kuang et al., “Knowledge-aware Alert Aggregation in Large-scale Cloud Systems: a Hybrid Approach,” arXiv:2403.06485, 2024.

[13] W. Zhang et al., “A Survey of AIOps for Failure Management in the Era of Large Language Models,” arXiv:2406.11213, 2024.

[14] European Union, “Regulation (EU) 2022/2554 (Digital Operational Resilience Act),” Official Journal of the European Union, 2022.

[15] H. Chen, P. Chen, and G. Yu, “A Framework of Virtual War Room and Matrix Sketch-Based Streaming Anomaly Detection for Microservice Systems,” IEEE Access, vol. 8, pp. 43413–43426, 2020, doi: 10.1109/ACCESS.2020.2977464.

[16] E. Kidmose, M. Stevanovic, S. Brandbyge, and J. M. Pedersen, “Featureless Discovery of Correlated and False Intrusion Alerts,” IEEE Access, 2020, doi: 10.1109/ACCESS.2020.3001374.

[17] L. Korzeniowski and K. Goczyla, “Landscape of Automated Log Analysis: A Systematic Literature Review and Mapping Study,” IEEE Access, vol. 10, pp. 21892–21913, 2022.

[18] S. Ahmed, M. Singh, B. Doherty, E. I. Ramlan, K. Harkin, M. Bucholc, and D. Coyle, “Knowledge-based Intelligent System for IT Incident DevOps,” in Proc. 2023 IEEE/ACM International Workshop on Cloud Intelligence & AIOps (AIOps), 2023, pp. 1–7, doi: 10.1109/AIOps59134.2023.00005.

[19] Kaggle, “Financial risk and fraud datasets (e.g., credit card fraud detection, loan default, and credit scoring),” accessed 2025-01. [Online]. Available: https://www.kaggle.com/datasets

[20] UCI Machine Learning Repository, “Credit scoring and related datasets,” accessed 2025-01. [Online]. Available: https://archive.ics.uci.edu

[21] Ashish Babubhai Sakariya (2018). Leveraging CRM Tools to Boost Marketing Efficiency in the Rubber Industry. International Journal of Scientific Research in Science, Engineering and Technology (IJSRSET), 4(11) 354-363.