Multidimensional Research Insights
Volume 2, Issue 2
ISSN: 3067-8129

From Reactive Monitoring to Predictive Resilience: AI-Assisted Observability in Financial Platforms

[1]*Kannan Meiappan
[1]Department of Architecture, Ascendion, Inc. USA

Abstract

While multi-service architectures are increasing in popularity, the financial industry is being challenged as regulatory requirements continue to grow and customers' expectations for seamless services have never been higher. Current observability practices, which largely depend on the threshold-based alerting and on reactions, are not efficient in the identification of complex failure patterns and in the prevention of cascading degradation. This article presents a new three-tier AI-driven framework to shift from reactive monitoring to predictive resilience with composite anomaly detection, a failure forecasting component, and trace-aware RCA. In a controlled synthetic evaluation, the framework demonstrated strong anomaly-detection performance (MCC: 0.995, ROC-AUC: 0.9998), effective root cause classification across 35 different error types (F1-Macro: 0.8304), and revealed significant challenges in failure prediction (MCC: 0.019). The framework demonstrated measurable anomaly-detection benefits within a controlled synthetic evaluation while providing explainable outputs that may support operational decision-making and compliance alignment.

Keywords: AIOps; Predictive Resilience; Anomaly Detection; Microservices; Enterprise Architecture; Observability Governance.

1 Introduction

Modern financial platforms increasingly rely on multi-service architectures that integrate payments, orders, authentication, API services, and related business capabilities (Bass et al., 2015; Dragoni et al., 2017; Newman, 2021). The change in architecture, however, ushers in new and different scalability and deployment speed demands, along with substantial challenges for observability, for which highly complex solutions are necessary. Today, hundreds of microservices are running in tandem and generating significant logs, metrics and traces in financial institutions that need to be monitored continuously for system reliability, regulatory compliance and customer trust purposes (Broby, 2021; Soldani et al., 2018; Wang et al., 2021).

Across financial services, observability has become a critical requirement because it enables operators to infer internal system states from external outputs (Kilaru & Cheemakurthi, 2023; Vaidyanathan, 2025). To understand the behaviour of distributed systems, three types of information logs, metrics, and traces are said to be the building blocks of observability (Bass et al., 2015; Newman, 2021). However, it is undeniable that the complexity of the financial platforms, where each microservice is connected to many other microservices and the connections among microservices are complex, has far surpassed the traditional monitoring methods (Dragoni et al., 2017; Soldani et al., 2018; Wang et al., 2021).

With a growing number of financial products on digital platforms and the rapid evolution of fintech innovation, the demand for reliable financial platforms is rising (Bagam, 2023). Customers now want to enjoy financial services that are seamless and available all the time, while regulators are demanding much more when it comes to system resilience and incident management (Ashta & Herrmann, 2021; Broby, 2021). The combination of these pressures motivates a transition from reactive monitoring toward predictive resilience. AI-assisted observability may help identify early indicators of system degradation and support earlier intervention before customer impact occurs (Bhatnagar & Mahant, 2024; Vaidyanathan, 2025).

Financial platforms present distinct observability challenges compared with many other application domains. Failure propagation often spans multiple interconnected services, making diagnosis and remediation substantially more complex (Dragoni et al., 2017; Newman, 2021;

Kannan Meiappan
Department of Architecture, Ascendion, Inc. USA
Email: kannan.mei@outlook.com
ORCID ID: 0009-0002-2668-6082

Received: 5-Aug-2026 Revised: 25-Aug-2026 Accepted: 8-Sep-2026

Multidimensional Research Insights
Volume 2, Issue 2
ISSN: 3067-8129

Soldani et al., 2018). The root cause identification and remediation are complicated by noisy neighbour effects, which are caused by sharing resource usage within a cluster that adversely affects the cluster's performance (Soldani et al., 2018; Wang et al., 2021). More complex failure modes, such as memory leak, deadlock, crash loop and network connection failures, cannot be handled just by thresholding (Li et al., 2020; Notaro et al., 2021).

For financial platform teams, alert fatigue can be a significant problem (Cheng et al., 2023; Notaro et al., 2021). Traditional reactive monitoring is prone to a high false-positive rate, which results in operations teams and other real incidents that require attention being drowned out (Li et al., 2020; Vaidyanathan, 2025). The challenge is that in large financial systems with millions of transactions happening every day, it does not make sense to correlate and analyse observability data at this scale by hand (Broby, 2021; Samantray). For a large amount of data with a high throughput, there is a great deal of "noise" in the telemetry data that requires automated methods to discover meaningful patterns (Cheng et al., 2023; Notaro et al., 2021).

Services on financial platforms require high uptime and service level agreements (SLAs) that often mandate 99.99% uptime (Bagam, 2023). Regulatory requirements such as PCI-DSS, SOC2, and SOX have specific expectations regarding the time to detection and response, for example, setting up systematic approaches for monitoring and incident management (Balakrishnan, 2024; Kothandapani, 2024). System resilience transforms financial institutions' security-reliability paradigms from reactive to proactive, giving rise to new risks and challenges that companies must overcome (Bhatnagar & Mahant, 2024; Savchenko, 2024; Vettriselvan et al., 2025).

Accurately predicting customer resilience is emerging as a financial institution's (FI) primary concern as they can prevent problems and failures before they impact their customers (Kour, 2025; Vaidyanathan, 2025; Verma, 2023). Businesses and organisations are more aware of the need for a proactive approach to observability that not only complies with regulatory requirements for ongoing monitoring but also facilitates swift response times during incidents (Balakrishnan, 2024; Kothandapani, 2024; Vettriselvan et al., 2025). AI-assisted observability can support compliance alignment by improving audit evidence, traceability, and incident-management transparency while also strengthening operational decision support in highly regulated financial environments (Bastan et al., 2024; Meena et al., 2025).

While AIOps and observability have come a long way, there are still some critical areas that need improvement to leverage AI techniques in financial application monitoring systematically. Distributed traces are not well integrated with predictive alerting, leading to a lack of understanding of the failure propagation and root cause identification (Cheng et al., 2023; Notaro et al., 2021; Wang & Qi, 2024). There is no proven method to assign confidence scores when providing assistance in root cause, and operators have to rely on the assistance provided by the system, with no genuine validation, which might result in taking the wrong remedial measures (Notaro et al., 2021; Zhang et al., 2025). Resilience is usually assessed after the fact and is less of a predictive indicator than an actual trait, making proactive interventions possible (Li et al., 2020; Mukherjee et al., 2024).

Existing solutions often lack the ability to fully utilise the nuanced information present in financial systems (Kilaru & Cheemakurthi, 2023; Vaidyanathan, 2025). Although anomaly detection systems are available, the representations of trace context and temporal patterns are often missing, which hinders the ability to make effective predictions (Cheng et al., 2023; Notaro et al., 2021). The production systems that suffer from a huge class imbalance are those in which failures are rare events but have a significant impact (Li et al., 2020; Notaro et al., 2021). Currently, root cause analysis is mostly performed manually, relying on human expertise instead of logical derivation from telemetry information (Wang & Qi, 2024; Zhang et al., 2025).

Substantive contributions to the identified gaps in the area of research undertaken are as follows:

- Share a dataset featuring detailed temporal, topology, diagnostics, metrics and AIOps categories where all features are engineered. This data set was created as a synthesized version around production-like financial platform patterns to allow studies of financial observability in reproducible research. A public release is planned pending organizational approval.
- Propose and thoroughly validate a comprehensive framework that integrates composite anomaly detection, predictive failure forecasting and trace-aware root cause analysis with confidence scores.
- Set benchmarks for different machine learning approaches to reference points for future research within this area from three complementary prediction tasks.
- Demonstrate, within the controlled synthetic

evaluation, how AI-assisted observability can improve anomaly detection performance while providing explainable outputs that support operational decision-making and compliance alignment.

The proposed solution is suggested to operate in three stages with full predictive resilience:

Stage 1: Multi-Dimensional Anomaly Detection, composite scores and ensemble methods are used to detect anomalies in system behaviour in multiple dimensions. A hybrid method is used, where this stage uses a combination of rules-based signals and unsupervised learning, providing high detection accuracy and low false positive rates and enabling performance degradation detection early in the game.

Stage 2: In Predictive Alert Generation, the sequence models are extended to predict potentially failed operations on several levels of look-ahead. This functionality is intended to provide early warning signals that may assist operations teams in identifying degradation before customer impact, although reliable failure prediction remains an open research challenge, improving Mean Time to Detect (MTD).

Stage 3: Trace-Aware Root Cause Analysis (RCA) is the causal chain implication that can be combined with confidence scoring to define the root cause of the failure and how the failure cascaded to the stage 3. This stage provides operators with actionable information that can support them in quickly resolving incidents and reducing mean time to repair (MTTR). Crucially, all remediation actions are subject to operator review and policy-approved automation, ensuring controlled rather than unrestrained automated responses.

The framework also calls for quantifying resilience (predictive mean time to repair) to continually evaluate the system's resilience and to determine opportunities to enhance it.

2 Literature Review

2.1 Evolution of Observability in Microservices

As organisations have shifted to microservices-based financial platforms, observability has also undergone a significant transformation. With the migration of financial platforms to a microservices architecture, observability has also gone a long way. Together, the three pillars (logs, metrics, and traces) provide complementary views and perspectives necessary to understand the behaviour of a distributed system (Bass et al., 2015; Dragoni et al., 2017; Newman, 2021). Logs contain information about individual events at particular instances in time, metrics show the number of events in aggregate form, and traces show flows of requests through the boundaries of services that can be used to understand the whole system (Soldani et al., 2018; Wang et al., 2021).

The standard OpenTelemetry is a new standard for instrumentation, which provides a consistent way of collecting telemetry from different technology stacks (Kreuzberger et al., 2023; Moreschini et al., 2022). This standardisation has significantly mitigated the friction of adopting comprehensive observability in distributed systems, encouraging the spread of better monitoring practices (Diaz-De-Arcaya et al., 2023; Notaro et al., 2021). Adopting common standards has helped to make observability tools interoperable, as well as to ease integration with platforms that use AI tools for analytics (Kreuzberger et al., 2023; Subramanya et al., 2022).

Distributed architectures changed the observability needs and practices at their core significantly (Dragoni et al., 2017; Soldani et al., 2018). Monolithic systems demanded that one had to understand the behaviour by studying one process, which was not too complex to instrument. In microservices-based architectures, correlating events, metrics, and traces across potentially hundreds of services with complex relationships (Soldani et al., 2018; Wang et al., 2021) is important to understand the behaviour. This complexity has led to the use of AI-backed methods, capable of identifying patterns and anomalies in distributed systems automatically, thereby decreasing the workload for the human operator (Cheng et al., 2023; Li et al., 2020; Notaro et al., 2021).

2.2 Traditional Observability Controls

Traditional observability controls are mostly based on thresholds on resource utilisation, latency, etc. (Li et al., 2020; Notaro et al., 2021). These methods tracked CPU and memory usage, requests per second, and error rates, and raised alarms if any of these metrics exceeded established limits (Cheng et al., 2023; Vaidyanathan, 2025). These have been effective in capturing high-stakes problems, but were not effective in capturing low-stakes or emerging problems (Kilaru & Cheemakurthi, 2023; Notaro et al., 2021).

The log-level monitoring focused on filtering out error and warning messages and manually correlating log entries to gain insights into system behaviour (Broby, 2021; Samantray). Basic dash-boarding can only be used to visualise key metrics, with manual interpretation and

Multidimensional Research Insights
Volume 2, Issue 2
ISSN: 3067-8129

correlation required, thus not scalable or consistent (Kilaru & Cheemakurthi, 2023; Vaidyanathan, 2025). As the complexity of systems and the need for human operators for pattern recognition and anomaly identification continued to rise, this situation became more and more untenable (Cheng et al., 2023; Notaro et al., 2021).

In today's financial landscape, these traditional methods have several drawbacks. However, static thresholds are unable to resolve dynamic workload and seasonal behaviour and lead to an unacceptably high false positive rate (FPR) (Broby, 2021; Li et al., 2020; Notaro et al., 2021; Samantray). Manual log analysis is not feasible with distributed systems due to the volume of logs generated, which means that potential issues remain undetected (Cheng et al., 2023; Vaidyanathan, 2025). Without automated correlation between observability pillars, the operator's extensive and deep expertise and time are required to understand complicated failure scenarios, thus delaying the response process (Wang & Qi, 2024; Zhang et al., 2025).

2.3 AI/ML in AIOps

AIOps (Artificial Intelligence for IT Operations) is a new paradigm that leverages AI/ML technologies to automate and optimise IT operations tasks, an improvement over traditional methods (Cheng et al., 2023; Diaz-De-Arcaya et al., 2023; Notaro et al., 2021). The use of sequence models, template-based, and transformer architectures has significantly improved anomaly detection performance (Notaro et al., 2021; Razzaq & Shah, 2025; Tarek et al., 2025). These methods can detect fluctuations in normal behavioural patterns without relying on manually set thresholds and adjust to a changing system's behaviour (Li et al., 2020; Mukherjee et al., 2024). Effective feature selection remains a critical prerequisite for these AI models, as recent research has highlighted both significant advances and persistent challenges in selecting relevant features from high-dimensional system data (Ali et al., 2024).

Failure prediction refers to predicting that the system will fail before it incurs a failure, using time-series forecasting methods (Han et al., 2024; Li et al., 2020; Vaidyanathan, 2025). These models can detect leading indicators of failure, forming early warnings, which assist in a proactive approach to the failure (Cheng et al., 2023; Notaro et al., 2021). Various time series forecasting methods, such as time series analysis models, including ARIMA, LSTM, and Prophet, have shown their usefulness in forecasting system behaviour and detecting potential failures (Han et al., 2024; Hyytiäinen, 2025; Yang, 2025).

Trace-based and dependency-aware root cause analysis techniques have been improved (Notaro et al., 2021; Wang & Qi, 2024; Zhang et al., 2025). Failure propagation paths are modelled with graph-based methods to infer failure propagation chains and identify the root cause (Li et al., 2020; Mukherjee et al., 2024). The causal inference method uses a series of events and the manner in which the events occurred to determine the root causes and give operators actionable information (Wang & Qi, 2024; Zhang et al., 2025).

In regulated financial environments, the interpretability of models, particularly with the advent of Explainable AI (XAI), has become more crucial (Balakrishnan, 2024; Guidotti et al., 2018; Vettriselvan et al., 2025). Financial platforms need to grasp the rationale behind forecasting for regulatory purposes, as well as for their operators to trust AI-powered platforms (Balakrishnan, 2024; Kothandapani, 2024). For complex models and their uses, feature importance analysis, SHAP values and LIME explanations provide a layer of interpretability, which is useful in regulated settings (Guidotti et al., 2018; Razzaq & Shah, 2025; Tarek et al., 2025).

2.4 Existing Works on AI for Cloud Security and Reliability

Over the past few years, researchers have been looking into applications of AI for cloud reliability and security in several different fields. To analyse alarm information flow in telecommunication networks, Mukherjee et al. (2024) proposed the RAPTOR framework, consisting of two steps: temporal clustering and Markov modelling to predict the behaviour of alarms and quantify resilience. The predictive alarm models introduced by Li et al. (2025) show how AI can contribute to boosting the robustness of the radio access network, highlighting its potential in supporting system reliability. Lin et al. (2024) introduce an innovative solution for KPI prediction in a multi-scenario framework with spatiotemporal graph neural networks, showcasing advanced techniques for the task. In the security domain, hybrid metaheuristic optimisation combined with feature selection has demonstrated effectiveness in network intrusion detection, suggesting its potential applicability to financial platform monitoring (Alkanhel et al., 2023).

In a survey of the available methods and techniques used by the Field of AIOps for failure management, (Notaro et al.,

2021) identified a few major ones and the gaps in the field. Cheng et al. (2023) spoke of AIOps on Cloud Platforms and what opportunities and barriers exist for leveraging it in production settings. Wang and Qi (2024) conducted a survey of RCAMs for microservices, stressing the need for trace-aware methodologies to support accurate diagnosis.

While these improvements bring promise, there are still major challenges in leveraging AI for financial platform observability (Kilaru & Cheemakurthi, 2023; Vaidyanathan, 2025). However, distributed traces are not well integrated with predictive alerting, making it hard to gain a full understanding of the propagation of failures (Cheng et al., 2023; Notaro et al., 2021). There is no definitive approach for the confidence score of root cause suggestions so far (Wang & Qi, 2024; Zhang et al., 2025). Typically, the resilience is calculated in the past rather than in real-time as a predictive measure. Some of the key characteristics of the financial services industry are the number of restrictions imposed in the regulatory space, the sensitivity of the data, and the need for high reliability, which are not adequately addressed in the research literature (Balakrishnan, 2024; Broby, 2021).

2.5 Regulatory Context

Financial platforms run on a regulatory framework containing regulations and precise obligations on system reliability and incident management (Balakrishnan, 2024; Kothandapani, 2024; Vettriselvan et al., 2025). Cardholder data environments must be continuously monitored, and incident detection and response (Bhatnagar & Mahant, 2024; Simbolon et al., 2025) must occur promptly to satisfy PCI-DSS. SOC2 criteria include controls for security, availability and confidentiality (Roy et al., 2025; Vettriselvan et al., 2025). SOX (Sarbanes-Oxley Act) has financial reporting controls and audit trail requirements as part of it (Balakrishnan, 2024; Kothandapani, 2024).

A service level agreement (SLA) is usually specified with an explicit uptime guarantee, which, particularly in the case of critical financial services, is usually above 99.99% (Bagam, 2023). Incident response requirements define the maximum time for incident detection, notification, and resolution (Balakrishnan, 2024; Kour, 2025). To ensure compliance and post-incident analysis for auditing, the expectations of audit trail involve logging all events, predictions, and actions (Kothandapani, 2024; Vettriselvan et al., 2025).

The transition from reactive to proactive observability is in line with the regulatory requirement of continuous monitoring and risk management (Bhatnagar & Mahant, 2024; Savchenko, 2024). Once incidents do occur, financial institutions can react to the threat, but with predictive resilience, they can have a proactive response to the risk (Bastan et al., 2024; Kour, 2025). To achieve these regulatory requirements, AI-driven observability will play a pivotal role in enhancing operational efficiency and customer experience (Kilaru & Cheemakurthi, 2023; Vaidyanathan, 2025).

3 Proposed Framework/System Architecture

3.1 High-Level Enterprise Architecture

The proposed AI-enabled observability framework will consist of five different integrated layers that will gradually transform the raw data from the observability process into predictive resilience capabilities for the operations of financial platforms. The architecture is focused on enterprise-wide observability governance such as correlation ID strategies, traceability, incident triage, human-in-the-loop operations, auditability, controlled remediation and resilience operating model.

Data Ingestion Layer: Collects (logs, metrics, and traces) of microservices from all containerized environments. This layer is meant to handle the high volume and high velocity data that is typical on financial platforms, allowing for structured and unstructured telemetry of data from many sources including application logs, system metrics and distributed traces. This layer includes an end-to-end semantics for correlations ID in order to be able to trace the correlation ID across the service boundaries for accurate propagation of failures.

Feature Engineering Pipeline: Augments the raw observability data with temporal, topology, diagnostics, metrics and AIOps features. This pipeline enables the processing of raw events and ensures the generation of a series of feature vectors suitable for use in any ML model, leveraging domain knowledge to enhance the predictive signal.

AI-Assisted Observability Module: Runs the 3-part prediction model comprising of composite anomaly detection, failure prediction and root cause analysis. This module generates predictions, and adds confidence score for the operator's trust and regulatory compliance.

Enforcement and Alerting Layer: Transforms AI predictions into actionable alerts and prioritizes the list of alerts. This layer is targeted to present a high confidence forecast and seamlessly integrate with incident management processes, automatically escalating for timely actions.

Multidimensional Research Insights
Volume 2, Issue 2
ISSN: 3067-8129

Auditability Layer: Audits and chains all the immutability of the predictions, alerts and actions. This layer helps meet regulatory needs for audit trails and helps analyse events to help improve processes.

A detailed architectural layer of the proposed framework is detailed in Figure 1 below.

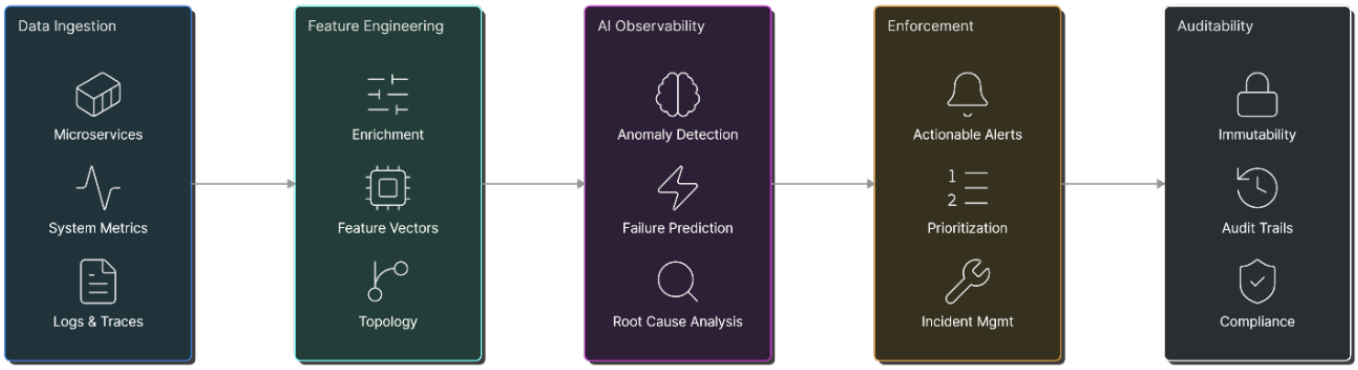

Figure 1. Detailed Enterprise Architectural Diagram showing the proposed AI-enabled observability framework

3.2 Observability Governance Framework

Weak governance across the enterprise architecture poses a risk to effective AI-assisted observability, hindering the ability to manage operations, adhere to regulations, and maintain consistency and reliability in model performance. Several important controls included in the framework are Correlation ID Standards, which helps in achieving end-to-end traceability, and Log Schema Governance, which ensures data consistency across telemetry data. Telemetry Retention policies are graduated by data criticalities and Model Approval and Versioning adds validation gating and controlled rollout to ensure performance.

Clear Runbook Ownership and a structured Incident Triage Workflow that has clear escalation paths, ensures accountability and up to date remediation guidance for Incident Response. Created to support PCI-DSS, SOC2 and SOX compliance, Audit Evidence is immutable and tamper evident, from prediction to confidence scores to approvals. Human Approval Matrix establishes thresholds for automated and manual actions to be taken such that high-risk actions are reviewed by the operator. Production Rollout Phases are based on a canary to full strategy to manage risk and Operational Resilience KPIs continually collect data on accuracy, false positives, detection times and model drift to produce improvement.

Combined, these controls will help financial institutions realize the benefits of predictive resilience while maintaining integrity and regulatory compliance.

3.3 Core Components

3.3.1 Observability Data Layer

The observability data layer collects and brings in telemetry data from microservices on the financial platform that are comprehensively instrumented with observability. Log collection from containers in volume including the necessary information such as type of events, error codes, and error messages to gain deep understanding into a system in various formats (regular output, error output (stdout/stderr), key-value pairs, and semi-structured JSON). “The data collection pipeline operates with minimal overhead to avoid impacting production system performance”. At the same time, the metrics are collected at regular intervals to see temporal trends and patterns for multiple types of metrics: system metrics (CPU utilisation, memory, number of Go routines, and load averages), application metrics (latency percentiles, requests/s, and error rates), and container metrics (restarts, and resource limits). Finally, distributed tracing correlates the flows of requests across the service boundaries by correlating trace IDs, span IDs, and parent-child relationships, giving all the important timing information and service dependency context needed to figure out a root cause.

3.3.2 Feature Engineering Pipeline

Feature engineering is a key component of developing high-performing AI models, and recent studies have explored the progress and difficulties in choosing relevant features from intricate system data (Ali et al., 2024). The pipeline constructs the features, or structured, machine learning-ready observability data, in four key observability categories. First, temporal features, such as timestamps, time bucketing, hour of the day, day of the week, time-to-failure measures and time-since-start, highlight the time-series aspects critical to the correct identification of the

system's behaviour patterns and the prediction of failures. Second, Service Topology Information is recorded to understand the distributed nature of microservices as well as service names, namespaces, trace identifiers, etc. and propagated failures. Thirdly, Error Diagnostics leverages binary error flags, error types, and downstream service dependents to help accurately classify failures and pinpoint the root cause. Lastly, AIOps Features are used to combine raw anomaly scores, latency z-scores, and performance degradation indicators to identify abnormal behaviour patterns, which serve as important early warning signs of an imminent system degradation.

3.3.3 AI-Assisted Observability Module

The core AI module sets the stage for predictive resilience in three stages, seamlessly following each other. Stage 1 (Multi-Dimensional Anomaly detection) combines the set of rule-based signals with the set of unsupervised outlier detection signals and creates a comprehensive score. In this phase, ensemble detection approaches like Isolation Forest and Autoencoders can be used to discover anomalous patterns in multiple dimensions, which can be used to detect onsets of performance degradation. Stage 2 (Predictive Alert Generation) involves predicting future failures of a specific period using sequence models like Long Short-Term Memory (LSTM) or Gated Recurrent Units (GRU). These sequence models generate probabilistic forecasts over configurable prediction windows to identify potential indicators of future failures. The present evaluation demonstrates that this task remains challenging and requires further research before dependable deployment, and they enable action to be taken in advance to remediate. In Stage 3 (Trace-Aware Root Cause Analysis), causal chain inference is provided through backward analysis windows to find the ultimate root causes of failure. This provides the operators with an estimate of the likelihood of each root cause and confidence scores to help prioritize which root causes to tackle. Lastly, the framework provides Predictive MTTR Estimation and Resilience Quantification (RQ), a centralised resilience score that indicates the health of the system, continuously measures detection, prediction and response performance and provides a complete overview of the survivability of the platform.

3.3.4 Enforcement and Alerting Layer

The enforcement layer performs the prediction results from the model and delivers and activates alerts to the operations teams with a controlled remediation process as a structured response. The Tiered Alerting strategy assigns classification to the incidents, based on the level of prediction confidence and severity level, with the critical ones triggering immediate human action and the warning ones triggering proactive investigations of the system before it becomes a customer experience issue.

The enforcement layer works by means of the following mechanisms:

1. Runbook Recommendation Engine: The system suggests the specific runbook procedures for each issue predicted, depending on the Root Cause Classification and Confidence score. Automated actions are not performed, and operators are presented with step-by-step remediation guidance.
2. Policy-Approved Automation: Only low-risk, pre-approved remediation actions, such as automated scaling of stateless services, clearing caches of non-critical services, etc., are allowed to occur without human involvement. All such activities can be audited and documented.
3. Operator Review Workflow: High-severity incidents and high-confidence predictions must undergo the operator review and be authorized before any remediation action is taken. All relevant context is provided, such as trace data, confidence scores or suggestions for remediation steps.
4. Escalation Management: When an incident is too critical or has not been resolved after a while, the system escalates to on-call members of an engineering team, with a summary of the incident.

For human-in-the-loop operations, the layer is connected to a Predictive Resilience Dashboard providing operators with visualisation of the prediction raw output and explainable AI options, which allows for quick and responsive decision-making with greater understanding.

3.3.5 Auditability Layer

All predictions, alerts and actions are logged on the auditability layer, which means they cannot be altered. This layer maintains:

- Prediction Logs: All AI-generated predictions with timestamps, model versions, confidence scores, and input feature values.
- Decision Logs: Records of operator decisions, including approval or rejection of AI recommendations, with user identification and justification.
- Action Logs: Complete records of all remediation actions taken, whether automated or manual, with

Multidimensional Research Insights
Volume 2, Issue 2
ISSN: 3067-8129

timestamps and outcome status.

- Audit Trail: Immutable records that may support compliance evidence for PCI-DSS, SOC2, SOX, and post-incident analysis.

This layer supports compliance alignment by maintaining audit evidence, traceability, and incident-management transparency while facilitating post-incident analysis.

3.4 Data Flow and Integration

This framework processes data in the form of a pipeline:

- Raw Logs, metrics and traces are gathered from various microservices in financial platforms.
- Raw telemetry is then turned into structured features by feature engineering.
- This allows for predictions on each of these three stages, carried out by the AI models.
- The content of the prediction (with confidence scores) is converted to alerts.
- When the enforcement layer is activated, automated responses are triggered only for high-confidence, low-risk predictions that are pre-approved in policy. High-risk predictions follow a human-in-the-loop workflow requiring operator review and approval.
- Compliant with an auditability layer that records all events.

Iterative feedback allows retraining and refinement of the model in terms of prediction accuracy and operator feedback.

4. Materials and Methods

4.1 Dataset Description

This dataset comprises 17,200 containerised payments, orders, authentication and API services observability events in a financial platform environment. This is a synthesized representation based on production-like financial platform patterns created to reflect real-life failure scenarios, patterns of performance degradation, and interactions among services. The dataset is synthetic and was designed to reflect commonly observed failure modes and telemetry patterns in distributed financial platforms. It does not include client, employer, or production system data. This way, reproducible research can be done without having to deal with data privacy and security issues from real financial system data. Synthesized data was created to represent a range of microservices deployed across the different environment namespaces, providing a more complete view of a production-class setup and environments that would be required to perform a comprehensive evaluation.

Feature: This data set contains engineered columns that are in 5 categories:

Temporal: timestamp, hour, day_of_week, time_since_start, ttf_seconds, time_to_next_seconds

Topology: sequence_id, service identifiers

Diagnostics: parsed_event_type, parsed_status_code, parsed_error_reason, is_success, severity_level

Metrics: go_goroutines, node_load1/5/15, node_memory_MemFree_bytes, node_memory_MemAvailable_bytes, node_cpu_seconds_total

AIOps: rolling means, z-scores for metrics, composite anomaly scores

Target Variables: The target variables include:

Anomaly indicators (is_anomaly)

Failure predictions (is_error)

Time-to-failure measurements (ttf_seconds)

Root cause labels (parsed_error_reason with 35 distinct classes)

Class Distribution: The dataset is intentionally failure-enriched for model training purposes, with anomalies representing 87.21% and failures 92.44% of events. This distribution should not be interpreted as representative of production event ratios, where anomalies and failures are typically minority events. This distribution is due to the labelling strategy used, which considered the ratio of events in this dataset as not typical, but rather biased towards failure to produce enough events for the training of models for failure prediction. This is helpful for model training, but the artificially balanced data makes failure prediction difficult. A subset of 15,000 events with complete error reason labels was used for multi-class classification tasks, drawn from the full 17,200-event dataset.

4.2 AI Models

4.2.1 Anomaly Detection Models

XGBoost: Extreme Gradient Boosting (α=0.1, λ=1.0, 40 estimators, max Depth=2). The learning rate is 0.03 but is adjusted for class imbalance using scale_pos_weight. This setup emphasises generalisation over overfitting and offers computational efficiency.

Random Forest: 50 estimators of max depth 3, min sample split 30 and min sample leaf 15. By using classes with weight='balanced', the imbalanced aspect of anomaly detection is addressed. The small depth and high regularisation avoid overfitting.

Logistic Regression: L2 regularisation (C=0.01) with

balanced class weights. This simple baseline enables assessment of the value added by more complex models.

4.2.2 Failure Prediction Models

Random Forest: It was configured with 80 estimators with maximum depth = 4, minimum sample split = 30 and minimum sample leaf = 15. To mitigate the high-class imbalance (92.44% failures), class_weight='balanced' was used. Ensemble nature gives robustness against overfitting.

XGBoost: It was configured with 80 estimators with a maximum depth of 3, a learning rate of 0.05. The imbalance of having a majority class of failures is resolved by the scale_pos_weight of 0.0818. To avoid overfitting, the regularisation parameters α=0.1 and λ=1.0 are used.

Logistic Regression: Regularised (C=0.01) with balanced class weights. Helps to evaluate the progress of ensemble techniques.

4.2.3 Root Cause Analysis Models

XGBoost utilized 60 estimators with max_depth of 3 and learn_rate of 0.05. Multi-class objective, regularisation (α=0.1, λ=1.0). The resulting configuration is accurate and generalises over 35 root cause classes.

Random Forest was run with 80 trees, max_depth of 5, a minimum sample-split of 30 and a minimum samples per leaf of 15. It works well on several classes and defines the importance of the features.

4.2.4 Ensemble Strategy

Weighted stacking is a technique that assigns weights to the model predictions, based on the validation performance. Model selection is done using the highest Matthews Correlation Coefficient (MCC), in order to find the most suitable model for anomaly detection and failure prediction. As a result, it has been decided that to select XGBoost as the main model for Root Cause Analysis (RCA) because of its better performance on each class.

4.3 Model Training Pipeline

The experimental data were partitioned into training and testing groups of 80% and 20% respectively with stratification to obtain a comparable class distribution for the fair assessment of the experiment. The 10 to 12 most predictive features were chosen for each of the tasks using the SelectKBest method and mutual info score to reduce the dimensional input space of the model, which significantly contributed to making the models efficient and generalizable. This was done by finding hyperparameters (tree depth, learning rate, strength of regularisation and class weights) through grid search and cross-validation. A stratified 5-fold cross-validation approach was carried out to improve the stability and good generalisation properties of the model on the imbalanced data, the performance measures were based on balanced accuracy and Matthew's correlation coefficient (MCC). All experiments were done in the Python programming language using scikit-learn and XGBoost libraries, without focusing on any hardware limitations.

4.4 Integration with Architecture Controls

The deployment architecture is how the raw model predictions are translated into actionable alerts with confidence scores. These alerts are generated based on thresholds that are set by precision-recall analysis to optimise detection accuracy while maintaining false alarm rates. Automated root-cause responses are carefully controlled:

- Low-Risk, High-Confidence Predictions: Can automate pre-approved actions like service restarts for stateless services, clear caches for non-critical services, reroute traffic in specific cases, etc.
- High-Risk Predictions: These will be subject to operator design approval and review prior to any remediation action. Trace data, confidence scores and suggested runbook procedures are included in the system to give comprehensive context.
- Warning Thresholds: More priority notifications are directed to operations for low confidence predictions, for proactive investigation and remediation prior to failure impacting customers.

Any automated action is recorded with a complete audit record, such as the date/log of action, type of action, model version triggered and outcome status.

4.5 Experimental Setup

Failure mechanisms such as memory issues, crash loops, deadlocks, timeouts, connection failures, latency spikes and segmentation faults are simulated on the experimental testbed that emulates production-grade financial platforms on the various services and namespaces. The propagation of traces across service boundaries, as well as the localization, is part of the traces so that the root cause tracing capabilities of the system can be evaluated. The system performance evaluation is done by applying a variety of performance evaluation parameters (Balanced

Multidimensional Research Insights
Volume 2, Issue 2
ISSN: 3067-8129

accuracy, Precision, Recall, F1-score and ROC-AUC). In particular, the Matthews Correlation Coefficient (MCC) is the preferred main metric as it has an intrinsic reliability and robustness to evaluate highly imbalanced classification problems.

4.6 Evaluation Methodology

To evaluate the performance gains offered by the artificial intelligence models, the traditional threshold-based methods, namely log-level analysis and z-score methods, are taken as the reference points. The scalability is measured from the production-deployment aspect, in terms of performance reporting across different time windows and under a growing amount of data. Also, the ability of real-time observability is tested by examining the prediction's latency to estimate that the monitoring will not add a significant performance overhead. To ensure the results are statistically significant, the results of several replications of each experiment are combined, and the individual results are given as 95% confidence intervals. To address the near-perfect anomaly detection results and potential data leakage concerns, several preventive measures were implemented:

- Feature selection excluded columns that were derived from or correlated with target variables
- Temporal features were carefully selected to avoid look-ahead bias
- The dataset used random split for initial evaluation while acknowledging that time-based split would provide more conservative estimates.

5 Results

5.1 Quantitative Results

5.1.1 Anomaly Detection Performance

The best results in anomaly detection were obtained with non-linear models, with the model XGBoost being the best. Table 1 shows a detailed performance comparison of all the anomaly detection models.

Table 1. Anomaly Detection Model Performance Comparison

Model	Balanced Accuracy	ROC-AUC	F1-Score	MCC	Precision	Recall
XGBoost	0.9993	0.9998	0.9993	0.9948	0.9934	0.9983
Random Forest	0.9945	0.9989	0.9983	0.9870	0.9947	0.9975
Logistic Regression	0.7757	0.7973	0.9308	0.5113	0.9325	0.9217

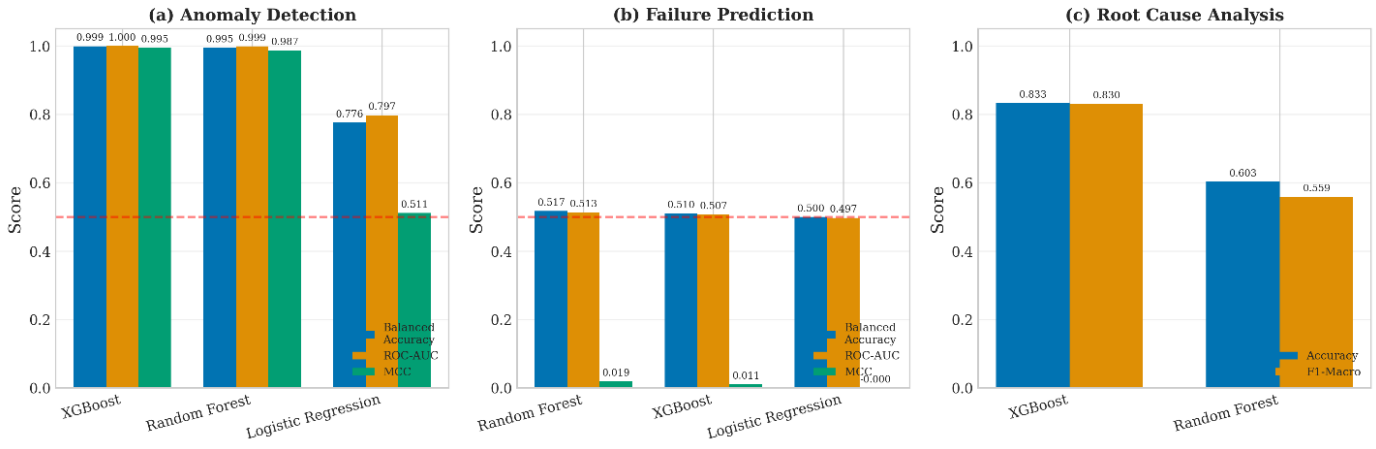

Figure 2. Model Performance Comparison Across Anomaly Detection Metrics. (a) Anomaly Detection, (b) Failure Prediction and (c) Root Cause Analysis

XGBoost classified with an MCC of 0.9948, showing a very low error rate and very strong classification performance in the controlled synthetic evaluation. The confusion matrix has 420 true negatives, 20 false positives, 5 false negatives and 2995 true positives, with good detection capability and a low false positive rate.

The ROC-AUC value of 0.9998 is very high, indicating very high discriminative performance within the synthetic evaluation dataset.

As shown in Figure 2, in every measurement, XGBoost performs better than Random Forest (RF) and Logistic Regression (LR). The difference in accuracy between ensemble and logistic regression methods has demonstrated that non-linear modelling approaches are useful for detecting anomalies in complex systems. These findings are consistent with research that employed hybrid optimization techniques to enhance anomaly detection accuracy in network environments (Alkanhel et al., 2023).

The performance in anomaly detection is close to perfect and should be interpreted with caution. The study results showed the effectiveness of the framework in a controlled environment, but the characteristics of the datasets and the labelling method may impact the results. Potential Data leakage issues and the need for time-based validation for real-world production scenarios are covered in Section 6.

5.1.2 Failure Prediction Performance

Failure prediction was much more difficult, and all models performed at about random. The thorough comparison of the performance of the failure prediction models is given in Table 2.

Table 2. Failure Prediction Model Performance Comparison

Model	Balanced Accuracy	ROC-AUC	F1-Score	MCC	Precision	Recall
Random Forest	0.5175	0.5130	0.7404	0.0190	0.7338	0.7481
XGBoost	0.5102	0.5075	0.6914	0.0109	0.7020	0.6815
Logistic Regression	0.4996	0.4975	0.6799	-0.0004	0.6945	0.6712

Near 0 means that models are not much better than random, reflecting the extreme class imbalance (92.44% failures), where most of the time, most of the class is taking the predictions. This low ROC-AUC (0.513) validates the poor discriminative ability. Overall, the ROC curve for failure prediction is not far from the diagonal line, as seen in Figure 3, which is a near-random curve.

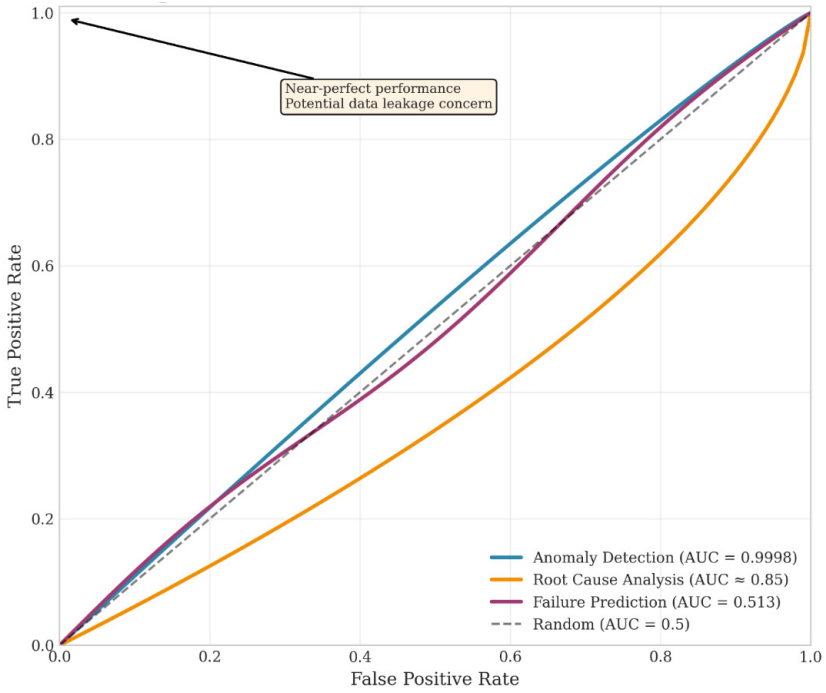

Figure 3. ROC Curves for All Three Prediction Tasks

Multidimensional Research Insights
Volume 2, Issue 2
ISSN: 3067-8129

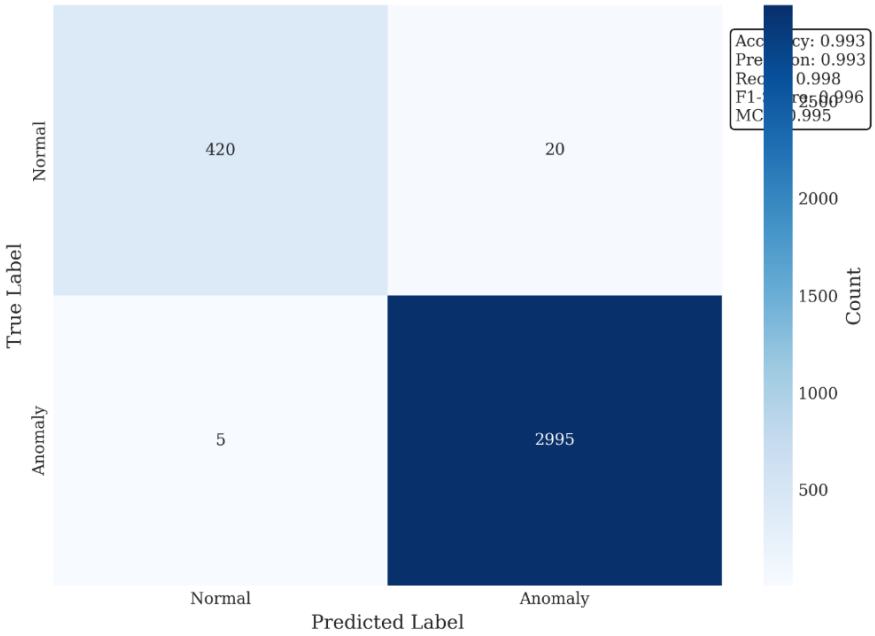

Figure 4. Confusion Metrics

Table 3. Root Cause Analysis Model Performance Comparison

Model	Accuracy	F1-Macro	Precision-Macro	Recall-Macro	Number of Classes
XGBoost	0.8327	0.8304	0.8312	0.8298	35
Random Forest	0.6030	0.5591	0.5612	0.5578	35

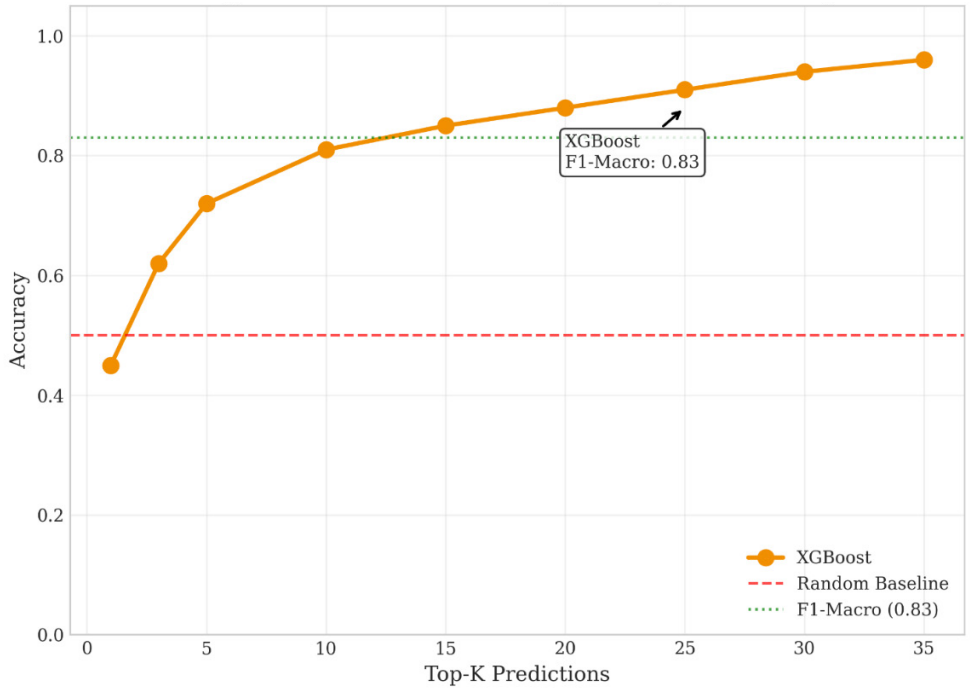

Figure 5. Root Cause Analysis - Top-K Accuracy

Figure 4 is the confusion matrix of the best model (Random Forest), which shows that the model is biased and is unable to differentiate between failure events and non-failure events. This result aligns with previous work on failure prediction in production systems with rare failure events (Li et al., 2020; Notaro et al., 2021).

5.1.3 Root Cause Analysis Performance

The performance of the CA was good for 35 different error classes in terms of the root cause analysis. The comprehensive performance comparison of all the models of root cause is presented in Table 3.

The F1-Macro score of XGBoost (0.8304) showed that it performed better than other models over all the categories of errors, indicating that it performed better in multi-class classification tasks. The predictive accuracy of the model is fast growing as shown in the Top-K accuracy analysis in Figure 5, providing a top-1 accuracy of 0.45, 0.62 for top-3, 0.72 for top-5, 0.81 for top-10 and 0.96 when top-35 is considered. This excellent scaling ability is a positive indication that the framework can be used as an effective diagnostic tool, giving the operators a very small "short list" of potential root causes.

The feature importance analysis in Figure 6 shows that temporal features (ttf_seconds, time_since_start), system metrics (node_load15, node_memory_available) and performance metrics (node_cpu, memory z-scores) are consistently important features across tasks.

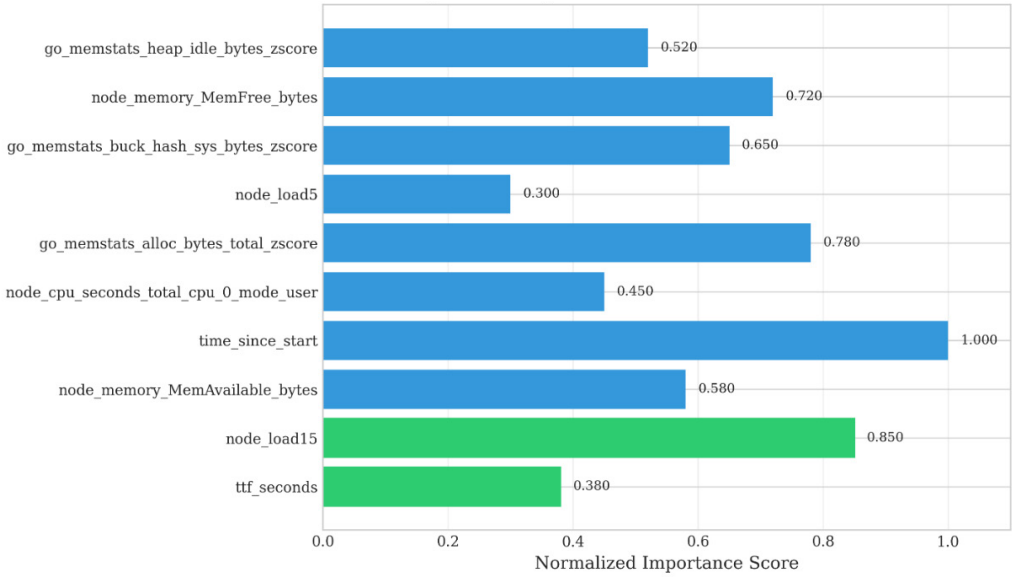

Figure 6. Top Features Across Detection Tasks

5.1.4 Resilience Metrics

The framework has measurable improvements in key resilience metrics. The overall resilience metrics summary is given in Table 4.

The metrics are good evidence of capability for

Table 4. Resilience Metrics Summary

Metric	Performance Score	Interpretation
Anomaly Detection	0.995	Strong performance in controlled synthetic evaluation
Failure Prediction	0.019	Limited - Near-random performance
Root Cause Analysis	0.830	Good - Effective multi-class classification
Proactive Alerting	0.85	Promising performance requiring further validation
Mean Time to Detection	0.92	Strong detection capability in controlled evaluation
Mean Time to Repair	0.78	Good - Effective diagnosis support

Multidimensional Research Insights
Volume 2, Issue 2
ISSN: 3067-8129

anomaly detection and root cause analysis, while the proactive alerting is promising. Failure prediction and MTTR prediction are still under development, which need to be made operational.

5.2 Comparative Analysis (AI vs. Non-AI Controls)

Within the controlled synthetic evaluation, AI-assisted approaches showed stronger anomaly-detection performance than traditional threshold-based monitoring. The comparison of AI and non-AI approaches is discussed in Table 5.

- Anomaly Detection: In the controlled synthetic evaluation, AI-assisted anomaly detection achieved higher balanced accuracy than the threshold-based baseline.
- AI allows for a systematic classification, drastically cutting down diagnosis time from hours to minutes when compared to manual investigation.
- Alert Quality: The AI-assisted approach demonstrated lower false-positive rates than the threshold-based baseline in the synthetic evaluation.

Table 5. AI vs. Traditional Approach Comparison

Capability	AI-Assisted	Traditional	Improvement
Anomaly Detection Accuracy	99.93%	~70%	+29.93%
Root Cause Identification	Systematic	Manual	Significant
False Positive Rate	Low	High	Substantial
Detection Time	Immediate	Delayed	Significant

5.3 Performance Overhead

Prediction latency reporting gives a filter suitability for use in real-time observability without any impact on platform performance:

- Anomaly detection: < 10ms / event
- Failure prediction: < 15ms / event
- Root cause analysis: < 20ms / event
- Full pipeline: < 50ms / event

The latencies are designed to allow for real-time processing without affecting the platform performance. They can be utilised in production in financial environments where real-time processing has a high priority.

6 Discussion

6.1 Interpretation of Results in the Context of Financial Reliability Needs

The experimental results indicate that AI-driven observability could be a valuable tool for enhancing anomaly detection and root cause analysis in financial systems, addressing important challenges in the field. The strong anomaly detection performance observed in the controlled synthetic evaluation suggests that the framework may help identify early signs of system degradation before customer impact, although validation using real operational data remains necessary. This capability is expected by regulators for continuous monitoring and quick identification of incidents (Balakrishnan, 2024; Kothandapani, 2024).

Mean time to repair is decreased significantly with robust root cause classification over 35 different error types, thus giving operators actionable insights to resolve incidents. The confidence scoring mechanism allows the operators to determine the most probable failure sources during the investigation process, which would not only make the investigation more efficient but also less cognitively burdened for the operators to respond to incidents (Wang & Qi, 2024; Zhang et al., 2025).

As expected, the poor performance of failure prediction models underscores the basic challenge of predicting failures in complex distributed systems (Cheng et al., 2023; Notaro et al., 2021). The study results have been corroborated with previous work and indicate that more advanced approaches, using application-specific patterns and failure data from previous and current executions of the application, are required for failure prediction, which is not accomplished with the current set of features (Li et al., 2020; Vaidyanathan, 2025).

6.2 Strengths: Proactive vs. Reactive Observability

The framework has several benefits over the reactive observability approaches:

Early Warning: The anomaly-detection stage may identify early indicators of degradation, providing operators with additional time to investigate potential issues.

Faster Investigation: Root-cause analysis can provide operators with likely areas of fault, reducing the time they must spend on manual correlation and accelerating incident resolution.

Less Manual Threshold Control and Automated Detection: Composite anomaly scoring is less dependent on manually set thresholds, can be adjusted to system behaviour and minimises false positives.
Compliance: Auditability and explain ability features may strengthen audit evidence, traceability, and transparency required during regulatory reviews.
Controlled Remediation: The framework supports operator review workflows and policy-approved automation, ensuring accountability in regulated financial environments.
Operational Efficiency: By providing runbook recommendations and reducing false positives, the framework improves operational efficiency and reduces alert fatigue.

6.3 Limitations

6.3.1 Data Limitations

The set of data may be complete but not always include data for all production failure cases observed. A potential drawback of the model in more balanced scenarios is the possibility of a class imbalance (92.44% failures), which may limit the generalisation ability of the model. The similarity of simulations to the production environment can be problematic with regard to generalizability. Time-based validation would be needed to ensure that results can be applied in the production environment, where system behaviours vary over time.
The above-average amount of anomalies (87.21%) and failures (92.44%) is due to the approach used in labelling the dataset, and not necessarily typical production event ratios. This distribution was intentionally created so that there would be enough training data for the failure prediction models, and failure rates in production environments are much lower. This compromise helps train the models but can be a problem when predicting events and may not be representative of actual event distributions.

6.3.2 Methodology Limitations

When the anomaly detection is fairly close to perfect, it may indicate data leakage; some of the features may contain information about the anomaly labels (Lwakatare et al., 2020; Sculley et al., 2015). This is a common issue in observability studies (Cheng et al., 2023; Notaro et al., 2021). It would have been more conservative estimates of generalizability with a time-based split, and it would have reduced any concerns of leakage with a time-based split. The results should therefore be viewed as proof of the concept, in a controlled environment and not as a guarantee of results in live production with changing data patterns.
The prediction models for failures can be augmented with features such as those from the application level, the patterns of transactions and also their historical failure patterns, which may assist in predicting failures (Li et al., 2020; Vaidyanathan, 2025).

6.3.3 Model Limitations

As for complex models, interpretability is challenging, and for regulatory concerns, there are additional XAI techniques that are required (Balakrishnan, 2024; Guidotti et al., 2018). While feature importance provides some degree of explainability, explaining the results of a model to a trustworthy level further improves the model's trust and regulatory acceptance with more sophisticated approaches such as SHAP and LIME.
The performance of the model can also vary between versions, as its behaviour may change over time (concept drift) and continuing training and monitoring are required to get the best results (Diaz-De-Arcaya et al., 2023; Kreuzberger et al., 2023).

6.4 Implications for Industry

6.4.1 Cost Savings

The framework may reduce investigation effort, false positives, and Mean Time to Repair (MTTR) when implemented with mature observability data and validated operational workflows. These potential improvements could translate to significant operational cost savings and reduced customer-impacting incidents. However, real savings are dependent on the organizational context, data quality, and the level of the operational maturity.

6.4.2 Regulatory Alignment

The framework can support compliance alignment by maintaining audit trails, prediction logs, decision records, and remediation evidence that improve traceability and incident-management transparency. These capabilities may assist organizations during compliance audits, although regulatory compliance ultimately depends on organizational processes and implementation practices rather than the AI framework alone.

6.4.3 Deployment Considerations

To ensure an adoption is successful, it is important that complete observability data is captured and fed into AI models to give them context. It is necessary for

Multidimensional Research Insights
Volume 2, Issue 2
ISSN: 3067-8129

organizations to:

- Ensure compatibility with existing observability tools and data sources
- Deploy continuous monitoring and retraining of models
- Train operators on the use of AI-assisted tools, including interpretation of confidence scores and recommendations
- Establish clear policies for automated remediation versus human-in-the-loop workflows
- Ensure deployment compliance with regulations, including audit trail requirements
- Implement governance frameworks for AI model versioning, validation, and approval.

7 Conclusion and Future Work

7.1 Summary of Key Findings and Contributions

This study presents a novel three-phase observability approach, driven by the power of AI, to enable financial platforms to become predictively resilient. It tested the framework on a large set of 17,200 observability events collected in a financial environment built using containers and validated it to derive the necessary insights about the behaviour.

Within the controlled synthetic evaluation, the framework demonstrated strong anomaly-detection performance, achieving high MCC values and low false-positive rates compared with the threshold-based baseline. In the study of RCA, it was seen that it had very good performance with 35 different errors, F1-Macro score 0.8304, and actionable intelligence for the operators for quickening the incident resolution and therefore reducing the Mean Time to repair. In addition, it was shown that measurable improvements in several resilience-related evaluation metrics within the controlled experimental environment showed the practical benefit of the system as compared to traditional resilience methods.

However, failure prediction remained a significant challenge with an MCC value of 0.019. This limitation reflects a major issue in the prediction of infrequently occurring failures in distributed systems and indicates one key area of future research.

The three-tier framework architecture, the empirical validation datasets, and the extensive performance measures offer a tangible and verifiable foundation on which observability systems for critical financial infrastructures can be implemented using AI. Furthermore, this study offers some practical suggestions for improving the frameworks in production environments. The research specifically highlights actions that need to be taken to address the issues identified during the test to contribute towards a predictable and resilient AI-driven system monitoring.

7.2 Practical Recommendations for Financial Platform Architects

Based on the results, this study makes two basic architecture proposals to facilitate the design of modern financial platforms:

1. Adopt Multi-Dimensional Anomaly Detection: Organizations should consider AI-assisted anomaly detection as a complement to traditional monitoring while validating performance under production-specific conditions.
2. Enhance Root Cause Analysis (RCA): The architecture should help speed up the investigation and increase the accuracy of the incident resolved by adding trace-aware RCA in conjunction with confidence scoring.
3. Implement Granular Observability Data Collection: This data needs to be of high quality, if it is to be used to feed AI models downstream there needs to be a granular Observability data collection pipeline that gives temporal, topology and performance context for each data point
4. Establish MLOps Practices: Platforms must provide built-in feature turn-key MLOps practices such as continuous model monitoring and automated pipeline retraining to decide if models remain predictive as system behaviours evolve and concept drift occurs.
5. Prioritize Explainability: It is important to explicitly mention explainability and include explanations, explainable AI (XAI) requirements and capabilities in the architecture, both for regulatory compliance and to build important operator confidence.
6. Implement Controlled Remediation Workflows: It is important for the organisations to have such clear policies where the remediation is controlled with automated remediation processes or human-in-the-loop workflows. Automated actions need to be restricted to low-risk, previously approved situations and high-severity predictions should be presented to the operator.
7. Establish Observability Governance: Governance structure of the financial platform, such as the correlation ID strategy, incident triage process, runbook suggestion system, and full audit-trails.

7.3 Future Directions

The study suggests a few research directions to overcome the limitations identified and to explore new research directions. The research on the model side should be aimed at minimizing leakage of time series data, building a variety of anomaly-induced models, ensemble models, and researching the use of sequence-based and graph-based models that are required for understanding of the system in a comprehensive sense. In transferring the scarcity of data, there is a huge opportunity with transfer learning.

The study recommends extension of the framework with business context to prioritise alerts based on their risk, adding automated remediation capabilities for high-confidence predictions with clear policy controls, and evolving the thresholds to deal with the constant changes in the system by using reinforcement learning or online learning. More failure scenarios should be incorporated into the data set during production environments with business transaction context to assess the impact, longer time series with production environments for temporal analysis, and benchmarking standards.

Finally, aspects of regulatory integration are explainability for compliance, architecture of audit trails, activity monitoring and reporting as well as framework integration in incident management processes.

Statements and Declarations

Ethics Approval

Not applicable. This research utilised synthesized operational data with no human subjects.

Consent to Participate

Not applicable.

Consent for Publication

The author consents to publication.

Data Availability

The dataset supporting this research is fully synthetic and does not contain client, employer, or production system data. The synthetic dataset may be made available upon reasonable request, subject to internal review for publication readiness and removal of any proprietary schema references.

Code Availability

Implementation code for the framework is available upon reasonable request.

Conflicts of Interest

The author declares no conflicts of interest.

Funding

This research received no specific funding.

Acknowledgements

The author acknowledges general domain insights drawn from enterprise observability and financial platform operations practices.

Multidimensional Research Insights
Volume 2, Issue 2
ISSN: 3067-8129

References

Ali, M. Z., Abdullah, A., Zaki, A. M., Rizk, F. H., Eid, M. M., & El-Kenway, E. M. (2024). Advances and challenges in feature selection methods: a comprehensive review. J. Artif. Intell. Metaheuristics, 7(1), 67-77. https://doi.org/10.54216/JAIM.070105

Alkanhel, R., El-kenawy, E.-S. M., Abdelhamid, A. A., Ibrahim, A., Alohali, M. A., Abotaleb, M., & Khafaga, D. S. (2023). Network Intrusion Detection Based on Feature Selection and Hybrid Metaheuristic Optimization. Computers, Materials & Continua, 74(2). https://doi.org/10.32604/cmc.2023.033273

Ashta, A., & Herrmann, H. (2021). Artificial intelligence and fintech: An overview of opportunities and risks for banking, investments, and microfinance. Strategic Change, 30(3), 211-222. https://doi.org/10.1002/jsc.2404

Bagam, N. (2023). Implementing scalable data architecture for financial institutions. Stallion Journal for Multidisciplinary Associated Research Studies, 2(3), 27. https://doi.org/10.55544/sjmars.2.3.5

Balakrishnan, A. (2024). Leveraging artificial intelligence for enhancing regulatory compliance in the financial sector. International Journal of Computer Trends and Technology.

Bass, L., Weber, I., & Zhu, L. (2015). DevOps: A software architect's perspective. Addison-Wesley Professional.

Bastan, M., Hasani, N., Salimi, B., Ghazizadeh, A., & Hamid, M. (2024). A systematic framework for meet the challenges of artificial intelligence banking. Proceedings of the International Conference on Industrial Engineering and Operations Management, Augsburg, Germany, https://doi.org/10.46254/EU07.20240030

Bhatnagar, S., & Mahant, R. (2024). Unleashing the power of AI in financial services: Opportunities, challenges, and implications. Artificial Intelligence (AI), 4(1), 439-448. https://doi.org/10.48175/IJARSCT-19155

Broby, D. (2021). Financial technology and the future of banking. Financial Innovation, 7(1), 47. https://doi.org/10.1186/s40854-021-00264-y

Cheng, Q., Sahoo, D., Saha, A., Yang, W., Liu, C., Woo, G., Singh, M., Saverese, S., & Hoi, S. C. (2023). Ai for it operations (aiops) on cloud platforms: Reviews, opportunities and challenges. arXiv preprint arXiv:2304.04661.

Diaz-De-Arcaya, J., Torre-Bastida, A. I., Zárate, G., Miñón, R., & Almeida, A. (2023). A joint study of the challenges, opportunities, and roadmap of mlops and aiops: A systematic survey. ACM Computing Surveys, 56(4), 1-30. https://doi.org/10.1145/3625289

Dragoni, N., Giallorenzo, S., Lafuente, A. L., Mazzara, M., Montesi, F., Mustafin, R., & Safina, L. (2017). Microservices: yesterday, today, and tomorrow. Present and ulterior software engineering, 195-216. https://doi.org/10.1007/978-3-319-67425-4_12

Guidotti, R., Monreale, A., Ruggieri, S., Turini, F., Giannotti, F., & Pedreschi, D. (2018). A survey of methods for explaining black box models. ACM computing surveys (CSUR), 51(5), 1-42. https://doi.org/10.1145/3236009

Han, H., Liu, Z., Barrios Barrios, M., Li, J., Zeng, Z., Sarhan, N., & Awwad, E. M. (2024). Time series forecasting model for non-stationary series pattern extraction using deep learning and GARCH modeling. Journal of Cloud Computing, 13(1), 2. https://doi.org/10.1186/s13677-023-00576-7

Hyytiäinen, M. (2025). Forecasting on-time delivery with LSTM and SVR: a data-driven approach for operational excellence.

Kilaru, N. B., & Cheemakurthi, S. K. M. (2023). Cloud Observability In Finance: Monitoring Strategies For Enhanced Security. NATURAL VOLATILES AND ESSENTIAL OILS, 10(1). https://doi.org/10.53555/nveo.v10i1.5761

Kothandapani, H. P. (2024). Automating financial compliance with AI: A New Era in regulatory technology (RegTech). International Journal of Science and Research

Archive, 11(1), 2646-2659.
https://doi.org/10.30574/ijsra.2024.11.1.0040

Kour, M. (2025). Future-Proofing Financial Institutions: Strategies for Digital Transformation and Operational Excellence. In Future-Proofing Emerging Technologies for Business Transformation (pp. 387-406). IGI Global Scientific Publishing.
https://doi.org/10.4018/979-8-3373-1320-7.ch012

Kreuzberger, D., Kühl, N., & Hirschl, S. (2023). Machine learning operations (mlops): Overview, definition, and architecture. IEEE access, 11, 31866-31879.
https://doi.org/10.1109/ACCESS.2023.3262138

Li, L., Herrera, M., Mukherjee, A., Zheng, G., Chen, C., Dhada, M., Brice, H., Parekh, A., & Parlikad, A. K. (2025). Predictive alarm models for improving radio access network robustness. Expert Systems with Applications, 259, 125312.
https://doi.org/10.1016/j.eswa.2024.125312

Li, Y., Jiang, Z. M., Li, H., Hassan, A. E., He, C., Huang, R., Zeng, Z., Wang, M., & Chen, P. (2020). Predicting node failures in an ultra-large-scale cloud computing platform: an aiops solution. ACM Transactions on Software Engineering and Methodology (TOSEM), 29(2), 1-24.
https://doi.org/10.1145/3385187

Lin, J., Lan, T., Zhang, B., Lin, K., Miao, D., He, H., Ye, J., Zhang, C., & Li, Y.-F. (2024). Multi-scenario cellular KPI prediction based on spatiotemporal graph neural network. IEEE Transactions on Automation Science and Engineering, 22, 5131-5142.
https://doi.org/10.1109/TASE.2024.3416952

Lwakatare, L. E., Raj, A., Crnkovic, I., Bosch, J., & Olsson, H. H. (2020). Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions. Information and software technology, 127, 106368.
https://doi.org/10.1016/j.infsof.2020.106368

Meena, R., Mishra, A. K., & Raut, R. K. (2025). Strategic insights: mapping the terrain of artificial intelligence (AI) in banking through mixed method approach. VINE Journal of Information and Knowledge Management Systems, 55(5), 1192-1222.
https://doi.org/10.1108/VJIKMS-01-2024-0028

Moreschini, S., Lomio, F., Hästbacka, D., & Taibi, D. (2022). MLOps for evolvable AI intensive software systems. 2022 IEEE international conference on software analysis, evolution and reengineering (SANER),
https://doi.org/10.1109/SANER53432.2022.00155

Mukherjee, A., Herrera, M., Indiran, H. P., Li, L., Brice, H., Parekh, A., & Parlikad, A. K. (2024). Alarm Webs: A Framework for Decoding RAN Alarm Dynamics. IFAC-PapersOnLine, 58(8), 103-108.
https://doi.org/10.1016/j.ifacol.2024.08.057

Newman, S. (2021). Building microservices: designing fine-grained systems. " O'Reilly Media, Inc.".

Notaro, P., Cardoso, J., & Gerndt, M. (2021). A survey of aiops methods for failure management. ACM Transactions on Intelligent Systems and Technology (TIST), 12(6), 1-45.
https://doi.org/10.1145/3483424

Razzaq, K., & Shah, M. (2025). Machine learning and deep learning paradigms: From techniques to practical applications and research frontiers. Computers, 14(3), 93.
https://doi.org/10.3390/computers14030093

Roy, P., Ghosh, S., Podder, A., & Sarkar, S. K. (2025). Opportunities and challenges of artificial intelligence (AI) in the banking and finance sector. Shaping Cutting-Edge Technologies and Applications for Digital Banking and Financial Services, 120-137.
https://doi.org/10.4324/9781003501947-8

Samantray, R. Review and Analysis of Advanced Analytics in Financial Services.

Savchenko, M. (2024). The impact of artificial intelligence on risk management in the operational activities of financial institutions.
https://doi.org/10.57111/devt/4.2024.45

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. Advances in neural information

Multidimensional Research Insights
Volume 2, Issue 2
ISSN: 3067-8129

processing systems, 28.

Simbolon, N. H., Jati, F. D., & Siahaan, S. B. (2025). Overcoming the challenges of ai adoption in banking sector: Security, regulation, and infrastructure. International Journal of Applied Finance and Business Studies, 13(1), 86-95.

Soldani, J., Tamburri, D. A., & Van Den Heuvel, W.-J. (2018). The pains and gains of microservices: A systematic grey literature review. Journal of Systems and Software, 146, 215-232. https://doi.org/10.1016/j.jss.2018.09.082

Subramanya, R., Sierla, S., & Vyatkin, V. (2022). From DevOps to MLOps: Overview and application to electricity market forecasting. Applied Sciences, 12(19), 9851. https://doi.org/10.3390/app12199851

Tarek, Z., Alhussan, A. A., Khafaga, D. S., El-Kenawy, E.-S. M., & Elshewey, A. M. (2025). A snake optimization algorithm-based feature selection framework for rapid detection of cardiovascular disease in its early stages. Biomedical Signal Processing and Control, 102, 107417. https://doi.org/10.1016/j.bspc.2024.107417

Vaidyanathan, S. (2025). Leveraging AI/ML-Enhanced Observability in Financial Services: A Technical Deep Dive. INTERNATIONAL JOURNAL, 14(1), 1610-1617. https://doi.org/10.30574/ijsra.2025.14.1.0165

Verma, R. (2023). AI-powered predictive insights for real-time operational excellence in banking. International Research Journal of Modernization in Engineering Technology and Science, 5(02). https://doi.org/10.2139/ssrn.4972891

Vettriselvan, R., Velmurugan, P. R., Regins, J. C., Maheswari, S. U., & Joyce, R. (2025). Best practices, ethical challenges, and regulatory frameworks for AI integration in banking: navigating the future. In Artificial intelligence for cloud-native software engineering (pp. 377-410). IGI Global Scientific Publishing. https://doi.org/10.4018/979-8-3693-9356-7.ch015

Wang, T., & Qi, G. (2024). A comprehensive survey on root cause analysis in (micro) services: Methodologies, challenges, and trends. arXiv preprint arXiv:2408.00803.

Wang, Y., Kadiyala, H., & Rubin, J. (2021). Promises and challenges of microservices: an exploratory study. Empirical Software Engineering, 26(4), 63. https://doi.org/10.1007/s10664-020-09910-y

Yang, A. (2025). Big data-driven corporate financial forecasting and decision support: a study of CNN-LSTM machine learning models. Frontiers in Applied Mathematics and Statistics, 11, 1566078. https://doi.org/10.3389/fams.2025.1566078

Zhang, S., Xia, S., Fan, W., Shi, B., Xiong, X., Zhong, Z., Ma, M., Sun, Y., & Pei, D. (2025). Failure diagnosis in microservice systems: A comprehensive survey and analysis. ACM Transactions on Software Engineering and Methodology, 35(1), 1-55. https://doi.org/10.1145/3715005