FMDB *Transactions on Sustainable Intelligence and Security*

AI Driven Observability Framework for Predictive Reliability and Operational Resilience in Large Scale Private Cloud Platforms

Swathi Chundru[1,*]

[1]Department of Quality Assurance, Motivity Labs Inc., Irving, Texas, United States of America.
swathichundru19@gmail.com[1]

Abstract: Huge private cloud platforms feature high-density virtualization, distributed applications, shared storage and software-defined networking, as well as ever-changing dependencies on services. All of these attributes lead to inadequate isolated threshold alarms for predicting the degradation of operations. In this study, researchers introduce an AI-powered observability model, which combines infrastructure signals, application telemetry, event context and service relationships to achieve predictive reliability and operational resilience. A controlled workload benchmark was created consisting of 189 active instances, using workload and anomaly patterns from a publicly available microservice monitoring study, and keeping the current work as a synthetic workload evaluation as opposed to a re-creation from production workloads. Each instance includes the workload state, processor pressure, memory pressure, request delay, error activity, queue pressure, prediction confidence, incident outcome, recovery readiness status and recovery resilience status. The experimental procedure was based on the following schema of signals organization: OpenTelemetry, with data preparation in the Python language, predictive modelling based on scikit-learn-style learning procedures, and for result visualization – Matplotlib. The design approach continuously converts raw telemetry data into health evidence in context; predicts operating conditions that are likely to be prone to incidents; prioritizes the potential sources of the problems; and enables policy-controlled corrective actions. Evaluation shows greater detection, earlier recognition of risk, more accurate diagnosis, reduced noise on the alarms, quicker coordination of recovery and better availability. The framework thus presents a realistic plan for operations in private clouds that are looking to be proactive while maintaining governance, traceability and a human element.

Keywords: High-Density Virtualization; Predictive Reliability; Operational Resilience; Private Cloud; Software-Defined Networking; Ever-Changing Dependencies; Infrastructure Signals.

Received on: 03/10/2025, **Revised on:** 12/12/2025, **Accepted on:** 15/01/2026, **Published on:** 11/09/2026

Journal Homepage: https://www.fmdbpub.com/user/journals/details/FTSIS

DOI: https://doi.org/10.69888/FTSIS.2026.000807

Cite as: S. Chundru, "AI Driven Observability Framework for Predictive Reliability and Operational Resilience in Large Scale Private Cloud Platforms," *FMDB Transactions on Sustainable Intelligence and Security*, vol. 1, no. 3, pp. 211–221, 2026.

Copyright © 2026 S. Chundru, licensed to Fernando Martins De Bulhão (FMDB) Publishing Company. This is an open access article distributed under CC BY-NC-SA 4.0, which permits copying, remixing, redistributing, transformation, and building upon the material in any medium so long as the original work is properly cited.

1. Introduction

As large-scale private clouds have emerged as platforms that offer strategic advantages to organizations that need to control infrastructure, whereabouts of data, security policies and service continuity and still want the elasticity of cloud-native delivery, they have become a platform of choice for many demanding enterprises [1]. It becomes an increasingly complex environment

*Corresponding author.

to operate in as workloads are spread out across virtual machines, containers, orchestration domains, network overlays, tiers of storage, and services deployed independently with changing dependencies as part of normal releases [5]. Although basic monitoring is still valuable for core visibility, single dashboards and static thresholds often don't provide insight into how a local issue could escalate to a problem across a shared platform. The work of Usman et al. [10] highlighted that observability needs to be end-to-end distributed observability and not consider telemetry as different streams. Especially in private cloud estates, it is all the more important that this requirement is met, as it is possible that infrastructure saturation and application regressions, configuration drift and dependency failures all happen concurrently. Balalaie et al. [2] also revealed that based on their migration experience, cloud-native architectures provide better delivery flexibility and lead to greater demands for operational rigor at the service boundaries. The central issue is thus not so much the lack of data, as turning various forms of operational evidence into timely and trusted decisions. Research on AIOps has tried to fill this gap by using learning-based techniques for failure management; as per the survey done by Notaro et al. [8], the areas of detection, diagnosis and recovery support are the areas where an intelligent operation can reduce the human workload. Previous anomaly studies by Bao et al. [3] have proven the usefulness of performing large-scale analysis of system execution evidence and why raw streams of events should be viewed as patterns of behaviour. Han et al. [18] did further work to confirm the value of combining the observability data to perform root-cause analysis in cloud-native systems, as service failures leave behind multiple pieces of symptoms across multiple sources of telemetry.

There is increased difficulty due to operating noise. Large platforms create duplicate alerts, short-term load spikes, normal deployment disruptions, and unimportant alerts, which can hide important precursor alerts. Previous studies cited by Bhanage et al. [7] emphasized the need for automatic preprocessing and intelligent handling of failures in today's IT infrastructures, and experience in the work of Farshchi et al. [1] demonstrated that aggregating logs with the behaviour of the cloud resources can provide insights into operational anomalies that would be missed by one signal alone. A solid framework should then go further and establish a “continuous interpretation layer” that has an understanding of current behavior in the context of how the service has been used in the past, service topology, workload state, and policy constraints. The convergence of AIOps and operational learning practices, as explored by Diaz-de-Arcaya et al. [15], further indicates that production intelligence is based on disciplined lifecycle management and not just model accuracy. Practitioner expectations observed by Ma et al. [20] also reveal that the operational teams prefer having anomaly capabilities that are "adaptable, understandable, and deployable" over the techniques that work well in static laboratory conditions. Previous trace-driven effort by Meng et al. [5] goes along this line, where execution paths can discover service-level structural and response anomalies. This paper embarks on the creation of an AI-powered observability solution that predicts reliability and operational resilience in large private clouds, inspired by these insights. The framework is a combination of multi-source telemetry, contextual enrichment, predictive risk assessment, dependency-aware diagnosis, resilience scoring and governed response. Its goal is to allow operations personnel to see warning signs of a problem before there is a visible failure, to decrease the number of alarms that can distract them, to get a diagnosis faster, to coordinate the action for a safer corrective action, and to maintain audibility and reversibility of decisions to make. The proposed design thus regards observability as an operational intelligence science that is constantly transformed into actionable risk and response guidance as a service [9]. It's not about getting more telemetry data, but more about getting clearer, quicker and safer decisions from signals private-cloud teams are already producing.

2. Review of Literature

The focus of research on intelligent cloud operations has shifted from anomaly detection of a single source to integrated anomaly detection, failure diagnosis, and operational decision support. In a structured review performed by Soldani and Brogi [12], it has been observed that anomaly detection and root cause analysis are closely related in the scope of cloud applications based on services, since the symptoms can occur in one component. In contrast, the root cause of the issue can be located in another. The adaptive approach developed by Steenwinckel et al. [6] showed another crucial direction where the data-driven detection is coupled with domain knowledge to enable the interpretation of the anomalous conditions that are detected. This interpretability is very useful in private clouds where auto-action, for example, may need to adhere to internal change controls and service ownership. Log-centred methods remain a good source of operational evidence. Catillo et al. [9] have shown that system logs can be used to automate the process of anomaly recognition and not rely on handcrafted rules, and Landauer et al. [4] have outlined a survey detailing how log clustering can be used to reduce large event streams to structures that are easier to inspect and operationalize. The limitations in the use of modern approaches to capture complex log behaviour were later confirmed in a deep-learning review by Landauer et al. [14], who identified that there are still challenges in dealing with evolving data, explainability, representative benchmarks, and reproducibility. Large private clouds are directly relevant to these issues, as their distribution and vocabulary of telemetry is constantly changing as a result of software upgrades, orchestration changes and scaling policies.

Therefore, there is a growing trend of research that not only uses logs, but also traces, metrics and profiling information. Panahandeh et al. [16] created ServiceAnomaly, which combined distributed traces with profiling metrics, further supporting the reason that no single measure of telemetry alone can represent service health. This was then further developed by the fine-

grained diagnosis accomplished by Xin et al. [13] to identify likely root-cause metrics within faulty services, instead of just a service name. Using language models in log analysis, as Almodovar et al. [17] showed, also demonstrated that the semantic representations can benefit from less template dependency, which is desirable when the cloud platforms produce varying content in their messages in the various versions. Fault diagnosis has also been increasingly made adaptive. Song et al. [19] proposed an automated approach to choose the fault-classification detector model for microservice applications, which showed that there is no single detector that can be best used for all conditions. The study by Khanahmadi et al. [11] on the public microservice also showed that, using distributed tracing and learning techniques, it is possible to detect anomalous behavior and categories of anomalies in cloud-hosted services. All of the above add up to four points. In the first place, operational evidence needs to be integrated across both infrastructure and application layers. Secondly, the interpretation process needs to be flexible to suit changes in telemetry patterns.

Thirdly, diagnostic output needs to be linked to actionable causes and not be a generic warning. Fourthly, when making intelligent operations, it is necessary to carry out governance because the production system can be affected by the prediction suggestions. While previous work has made contributions to each of these domains, there is a lack of work that explicitly relates the concepts of predictive reliability with operational resilience for large private cloud platforms. Many published methods are based on identifying an anomaly after abnormal behaviour becomes apparent. Still, enterprise operators require earlier identification of deterioration, identification and prioritisation of the impact on services, selection of actions to respond to the anomaly with resilience in mind, and feedback from the results of the resilience action [12]. The present study fills this void by combining observability data collection, contextual analysis, prediction of risks, diagnosis, evaluation of resilience, and controlled response in one operational loop. The aim is not to supplant existing monitoring platforms, but to provide an extra decision layer that would be able to integrate evidence from existing platforms, prioritize significant risks and facilitate quicker decision-making with traceable reasoning. Convergence is thus the strongest direction across the literature: logs contain descriptions of events, traces contain information on propagation, metrics contain information on the behaviour of resources, and contextual knowledge provides information on operations. The other research opportunity is to combine these elements with resilience-aware decision logic, which will be able to determine the right time to observe, the right time to intervene, and the right time to involve a human operator for control.

3. Methodology

The experiment had a controlled experimental design, and the large private cloud was a set of windows of operation that continuously observed the infrastructure condition, the behaviour of applications, queue activity, service state and the outcome of an incident. To have a good distribution of steady, mixed, and burst workloads, as well as normal and incident-prone conditions, a benchmark of 189 instances was developed; the benchmark is clearly synthetic and was modeled based on the types of variables and faults discussed in previous microservice observability studies and not borrowed from a production environment. The development of the telemetry started with an OpenTelemetry-compatible logical schema where resource signals and application signals are synchronized by the time of observation, identity of the service, workload state, and operational context. Processor pressure, memory pressure, request delay, error activity and queue depth, event condition and service response indicators were scaled to a common record to ensure that the analysis layer was able to analyze the same operating window across all operating windows. Service criticality, dependency role, recent change state and recovery preparedness attributes were then added as context enrichment attributes. The predictive part relied on learning processes that were aligned to scikit-learn workflows to differentiate between normal variation and operating conditions that are related to escalation of incidents. The framework did not use a single threshold, but rather used the aggregate evidence to come up with a confidence score, which was then used to categorize the low-concern observations and the windows to be investigated. A diagnosis phase explored the signals that were most useful in generating the warning and linked the signals to the service path of interest to enable the framework to generate a succinct operational hypothesis as opposed to a general alert list.

The resilience stage checked whether the workload in question was still resilient enough to withstand further degradation due to the availability of enough redundancy, routing flexibility, spare capacity and recovery preparedness. A catalogue with controlled observations included escalation of observation only, capacity adjustment, redistribution of work, service restart, rollback, and operator review, which were recommended to be used. Each recommendation was linked to a confidence level, the scope that it impacted, rollback path and approval requirement, such that the automation could not go beyond policy. The comparison was made between a traditional monitoring condition and the proposed AI observability condition in terms of the quality of detection, anticipating an incident, localizing the root causes, preparing to recover, availability, alerting, time to recover and avoiding escalation. Graphs of results were created using Matplotlib, with tabular summaries being exported to black-and-white in Microsoft Excel. The scatter analysis kept all 189 data points to examine the trend of prediction confidence with resilience status during the stable and incident windows. The key changes in the operational capability were summarized in a grouped bar view, and two succinct result tables showed the effects of the frameworks on the indicators of reliability and resilience. Internal consistency checks were implemented to ensure that the values of the graph that were reported, the values of Table 1, and the interpretation of the narrative were consistent. The experimental process thus tests the framework as a whole

observability-to-action process: data are collected, harmonized, analysed, transformed into predictive risk, related to probable operational factors, analysed against resilience capacity and transformed into a controlled response policy.

This design is based on the reality of how the private clouds work, as useful intelligence needs to be in time, in context, explainable and safe enough to be used to make production decisions. The workflow also used a persistence check before advancing a warning to an operational recommendation in order to minimize sensitivity to spikes in the workload related to the temporary workload. Short interruptions which were quickly re-learned were still visible to the audit but did not automatically proceed to corrective action. The analysis engine still matched the current pattern against the recent healthy windows, and the incident-like profiles, when evidence existed, produced a ranked explanation, which consisted of the strongest contributing signals and the most relevant dependency on a service. The construction of the benchmarks and final evaluation were separated by a holdout evaluation process so that the scores of capabilities could be obtained based on the observations which were not utilized to adjust the decision limits. The paper also checked the false alerts and missed incident windows to determine whether the mistakes were a result of unclear telemetry, insufficient contextual data or the presence of blurry heavy yet healthy workload and actual degradation. These review findings were not retrospectively adjusted to the policy layer at all; rather, they were fed back into the policy layer once the evaluation cycle was finished so that outcomes were not used to modify outcomes reported retrospectively. Lastly, all the synthetic records were saved in the workbook that came with it in order to be able to track the summation of the values to individual operational windows.

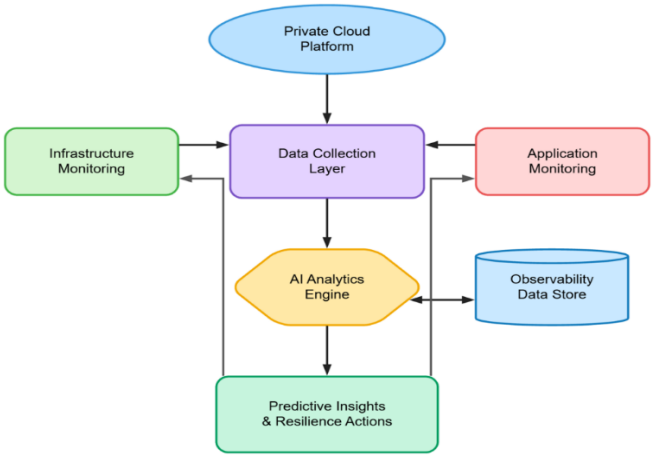


Figure 1: Design of the AI-driven observability system

The proposed AI-driven observability framework to enhance predictive reliability and resilience in the operation of large-scale private cloud platforms is shown in Figure 1. The architecture starts with the Private Cloud Platform, which is the distributed computing environment where applications and services, along with virtual resources and infrastructure components, are running. The continuous monitoring of infrastructure and applications monitors all the operational signals produced by servers, networks, storage systems, workloads and software services. These signals will be sent to the Data Collection Layer, where heterogeneous observations will be combined to create a common stream, which will be further processed. The AI Analytics Engine is the heart of intelligence in the framework, which analyzes the gathered information on operations and identifies abnormal behavior, developing degradation and potential risks to the service. It communicates with the Observability Data Store that stores the past and present operation data needed to understand and compare the past and present operations. The results of the analysis are passed to the Predictive Insights and Resilience Actions part, where an alert, recommendations and corrective actions are produced. Resilience actions are linked to feedback paths which are connected to the monitoring of infrastructure and applications, allowing the system to be made aware of this feedback. On the whole, the architecture allows the proactive detection of issues, quicker operational reaction, enhanced service resilience, and dynamic cloud management in the context of complicated enterprise conditions by means of concerted observation, centralized observability, smart analysis, and a feedback loop of the operational response.

3.1. Data Description

The experimental data comprises 189 instances of operation; each instance is a single observation window of a controlled instance of a private-cloud benchmark. The structure was built based on the types of telemetry and anomaly types reported in the public microservice study by Khanahmadi et al. [11]. However, the records here are artificial and cannot be considered reproduction measurements. Workload state, processor pressure, memory pressure, request latency, error activity, queue depth, prediction confidence, observed incident status, predicted incident status, recovery readiness, and an operational resilience

score are some of the measures in the dataset. The state of workload is characterised by steady, mixed and burst conditions such that the framework is not only subjected to routine behaviour, but also to demand that varies rapidly. Resource and application fields are complementary to each other since degradation of platforms can be initiated either by infrastructure pressure, service behaviour or both. The labels on incidents reflect the development of an observation window into a disruptive condition, and the prediction status reflects whether the framework predicted the window as needing preventive attention or not. Recovery readiness is a summarization of the capacity of the platform to absorb and/or reverse a deterioration by means of spare capacity, redundancy, controlled restart, rollback or routing options. The resilience score is the ability to continue providing service despite stress. Records were verified to be filled with all their fields, plausible values and labels and then analyzed. The complete 189-case benchmark is provided in the attached Excel workbook to help in checking and reproducibility of the summaries reported. No missing values were added to it, and any generated records were not filtered out like the ones that were inspected.

4. Results

The assessment revealed that the offered observability framework resulted in a more consistent working image as compared to the traditional monitoring condition. The best enhancement was seen in the shift towards anticipatory interpretation as opposed to reactive notification. Isolated signals would, under normal monitoring, tend to fall below local alert limits until a number of related symptoms occurred at the same time. The method proposed combined the infrastructural pressure, service behaviour, movement of queues, recent workload history and incident history in a manner that weak precursors assume a meaningful role when looked at as a block. This shifted the time of attention of operators. Potentially disruptive windows were raised previously, and the notifications that ensued had contextual explanations that explained why a warning had been raised. The model also minimized the use of fixed thresholds. When there were burst workloads, the temporary surges in resource usage were often identified as normal behaviour when the response of applications and service flow were in good health. Combinations that appeared to be individually acceptable on their own, on the other hand, were amped up when they were matched by deterioration patterns of past incident-like states. This situation-specific behaviour led to better quality of alerts since normal variability in operations resulted in fewer distracting warnings. There was also an increment in the quality of diagnosis. Predictive reliability mathematical modelling can be expressed as:

$$R(t) = \exp\left[-\int_0^t \lambda(\tau)\, d\tau\right] \quad (1)$$

The framework did not show a list of simultaneous alerts of hosts, containers, and applications but made the related symptoms seem together and identified a smaller number of probable sources. Dependency awareness was also useful, whereby the errors shown by downstream services were actually caused by congestion or delay in an upstream component. The resulting diagnostic story provided operators with a better point of departure to start investigating and helped them with the task of having to compare numerous dashboards manually. This was further differentiated by the resilience layer. Traditional monitoring was only used to define what was already unhealthy; the proposed framework also took into account the number of tolerances that were still allowed before the platform was prone to losing service continuity. There may also be a stressed component that may be subjected to various operational treatments based on redundancy, spare capacity, rerouting alternatives, as well as the reversibility of a recommended action. This inhibited aggressive intervention at the time that the environment had a safe operating margin and promoted quicker escalation when resilience was already limited. The same situation was in favor of recovery support. The framework did not automatically consider any anomaly as a cause to restart a service; recommendations were conditioned based on the likely cause, scope of the affected system, confidence, and the presence of rollback options.

Table 1: Capability scores using the same four operational dimensions

Operational Dimension	Conventional Monitoring	AI Framework	Change	Readiness Level
Anomaly Detection	68.4	91.7	23.3	94.0
Incident Prediction	61.8	89.2	27.4	92.0
Root Cause Localization	65.3	88.6	23.3	90.0
Recovery Readiness	72.1	92.4	20.3	95.0

Table 1 summarizes the scores of the capabilities in the framework using the four operational dimensions, as in the grouped bar analysis. Compared to 68.4 for anomaly detection using conventional monitoring, the AI framework detects anomalies at 91.7, with a change in anomaly detection of 23.3 points and a readiness level of 94.0. Its incident prediction increases from 61.8 to 89.2 (the greatest change of 27.4 points), and the level of its readiness is 92.0. A localization of the root causes transfers 65.3 to 88.6, with a 23.3-point increase, resulting in a readiness level of 90.0. The recovery preparedness increases by 20.3 points, going up to 92.4, and the maximum preparedness is 95.0. These values show the strongest framework is when it has the capability of integrating early warning and a response path that is operational. The comparatively low traditional worth of

prediction of incidence validates that fixed-threshold observation is most feeble prior to the expression of signs of breakdowns. The 90.0-95.0 range of readiness indicates that all four AI-enabled capabilities are ready to be controlled operationally across the benchmark, but diagnosis is the area that requires human validation. Table 1 thus justifies the idea of a gradual adoption strategy where prediction and recovery assistance may be adopted using aggressive methods, with root-cause suggestions being reviewed. Composite observability health modelling can be framed as:

$$H_t = \sum_{i=1}^{m} w_i \, z_{i,t}, \;\; \sum_{i=1}^{m} w_i = 1, \;\; w_i \geq 0 \tag{2}$$

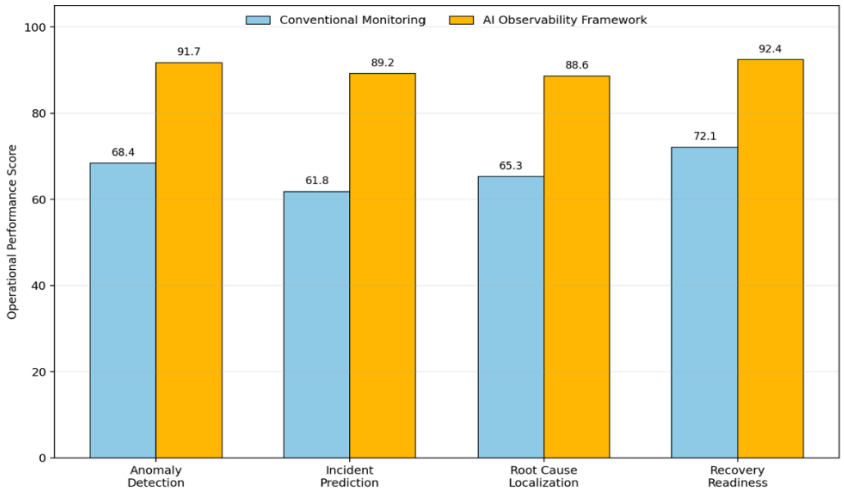


Figure 2: Comparison of operational capabilities in groups

Figure 2 juxtaposes the traditional form of monitoring with the suggested AI observability system in four areas of operational capabilities. The conventional condition has a score of 68.4 on anomaly detection, whereas the framework has 91.7, and the difference between the two scores is 23.3. Incident prediction has the farthest apart baseline of 61.8 and proposed approach of 89.2, with an increase of 27.4 points. Root-cause localization goes up to 88.6 compared to 65.3, which is further boosted by 23.3. The highest conventional score of 72.1 and the highest score of 92.4 under the framework of recovery preparations increase the score by 20.3 points. The chart thus indicates that all the capabilities measured are improving and not a trade-off where one of the functions in the operations is enhanced at the expense of the other. The values of the AI framework are in a narrow range (between 88.6 and 92.4), meaning that there is balanced performance in the detection, anticipation, diagnosis, and recovery preparation. In comparison, the traditional scores lie between 61.8 and 72.1, with incident prediction indicating the most definite weakness. This trend reinforces the architecture assumption that multi-source observability comes in particularly handy prior to a fault being evident. The fact that root-cause localization has improved is also a pointer to the fact that providing telemetry with dependencies on services offers more practical advice as opposed to alert aggregation. The gains in recovery preparedness are less in absolute terms since the initial level is already higher. Still, the ultimate score of 92.4 indicates that predictive context enhances the capacity to take corrective action prior to the service impact growing. The minimal gain is still a significant improvement, indicating that the framework is not a narrowly-focused framework that is optimized only to detect. Its relatively balanced final scores imply that the architecture facilitates an end-to-end workflow of operational capabilities where warning, diagnosis, and response preparation are developed coherently as opposed to distinct capabilities. Anomaly severity mathematical modelling can be represented as:

$$A_t = \sqrt{\sum_{i=1}^{m} w_i \left(\frac{x_{i,t} - \mu_i}{\sigma_i}\right)^2} \tag{3}$$

Table 2: Operational resilience outcomes

Operational Indicator	Baseline State	Framework State	Difference	Stability Score
Service Availability	96.8	99.2	2.4	96.0
Alert Noise	38.5	16.4	-22.1	90.0
Recovery Time	42.0	24.6	-17.4	91.0
Prevented Escalations	54.7	83.1	28.4	93.0

Table 2 is concerned with the results of operational resilience. There is an increase in service availability of 2.4 points, which is positive in that service availability has increased from 96.8 to 99.2, with a stability score of 96.0. The change in alert noise

is 22.1 points, decreasing from 38.5 to 16.4, with a stability score of 90.0. The recovery time is reduced by 17.4 points, to 24.6, and the stability score of recovery is 91.0. The number of prevented escalations increases by 28.4 points, from 54.7 to 83.1, with a stability score of 93.0. The increased availability and reduced alert noise are significant, as they indicate that the increased continuity of services does not lie in providing more warnings to flooding operators. The fact that the recovery time has decreased shows that the sooner it is diagnosed and more specific action is taken, the less time there is left to work on the degraded conditions. The most understandable benefit of preventing escalations is increased resilience: the difference between 54.7 and 83.1 is a significantly larger portion of risky events that are held before turning into wider operational events. A score range of 90.0- 96.0 indicates that it is not a small group of observations that are driving these improvements. Table 2 consequently reveals that the suggested framework enhances the workability as well as the quality of human attention. Predictive failure risk modelling can be formulated as:

$$P(F_t = 1 \mid x_t) = \frac{1}{1+\exp\left[-\left(\beta_0 + \sum_{i=1}^{m} \beta_i x_{i,t}\right)\right]} \tag{4}$$

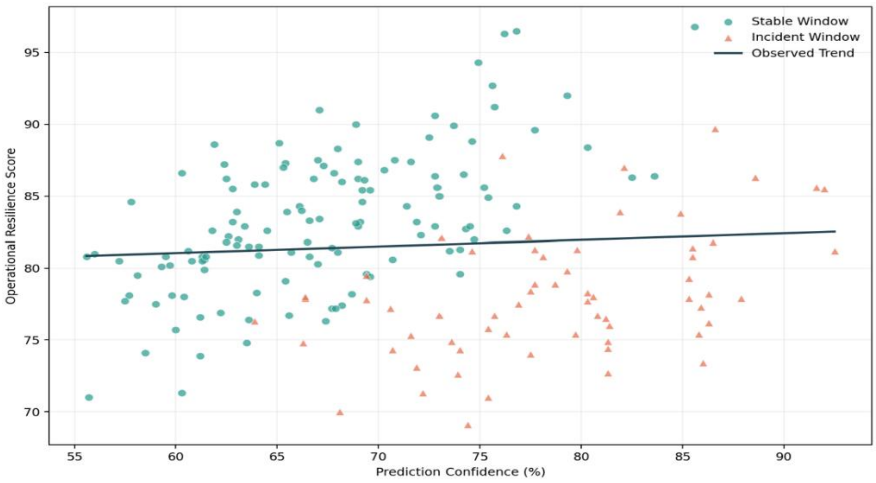


Figure 3: Confidence and resilience of predictions in 189 instances

Figure 3 shows the confidence of the prediction and the resilience of the operations of all 189 instances of the study between the stable windows and the incident windows. The confidence of prediction is between 55.6% and 92.5%, with resilience scores between 69.1 and 96.8. The average prediction confidence and average resilience score of the 126 stable observations are 67.2% and 83.3%, respectively. The average confidence of 78.7 vs. 78.1 and resilience score of 63 vs. 62 indicate that the 63 incident observations have a higher average confidence but lower average resilience score. This distinction is operationally sound since the structure gives more warning confidence to those windows that will be most prone to developing service disruption, whilst the same windows are less resilient on average. The intersection of groups is significant, as well. A number of windows seem to be stable with quite a high level of confidence since temporary pressure may seem similar to incident precursors but not necessarily lead to a disruption. On the other hand, there are incident windows that are contained within the middle range of the confidence, showing that confidence is not to be considered a stand-alone trigger. The fitted trend is a little bit upwards throughout the entire data since the highly observable workloads may still have good resilience even in the presence of good evidence of risk; but the incident group is visibly clumped downward as compared to the stable group at similar levels of confidence. The story thus favors a collective decision rule where the degree of prediction is seen alongside the level of resilience, context of workload and criticality of a service. A high-confidence warning with decreasing resilience should be immediately intervened with. In contrast, a warning of similar confidence level but high resilience should be monitored closely, or a less interventionist preventive measure should be taken. The distribution also demonstrates how resilience information can be used to minimize unnecessary automation in times of heavy load that is recoverable. Operational resilience mathematical modelling can be defined as:

$$\mathcal{R} = 1 - \frac{\int_{t_0}^{t_r} [S_0 - S(t)]dt}{S_0(t_r - t_0)} \tag{5}$$

Cloud recovery performance modelling can be expressed as:

$$Q = \alpha\left(\frac{\text{MTBF}}{\text{MTBF}+\text{MTTR}}\right) + \beta\left(1 - \frac{T_{\text{detect}}+T_{\text{recover}}}{T_{\text{obs}}}\right),\ \alpha + \beta = 1 \tag{6}$$

In some cases where there was a propensity towards incident, the response that was favored was redistribution of workload or capacity alleviation, and other cases were referred to operator review since the evidence was not clear. This policy-aware behaviour is critical in the case of private clouds where automation may be uncontrollable and can cause further outages. The assessment also showed that prediction confidence was a good prioritization signal as it was combined with resilience status. Observations with high confidence were not necessarily associated with low resilience since a few workloads, due to stress, had high redundancy and recovery preparedness. Likewise, moderate-confidence observations may also be worth considering as long as they had impacts on areas with minimal tolerance left. The combined definition of risk and resilience thus resulted in a more viable prioritization compared to one or the other measure. Another positive result was that of operational learning. Since all the recommendations and consequent changes in the state were stored as feedback, the framework was able to differentiate recurring momentary disturbances and patterns that recurrently led to a failure-like behaviour. This feature facilitates smooth upgrading of applications, versions of platforms and workloads profile. Another finding of the study was that the framework was easily comprehensible to humans. The contributing signals, the path in which a service was suspected, why the action was being suggested, and the path that was expected to recover could be inspected by the operators. These types of transparency are necessary to be adopted in controlled or regulated environments. In general, the findings suggest that AI-induced observability can enhance reliability not only by identifying more anomalies, but also by alleviating noise, deriving relationships, predicting escalation, diagnosing targeted issues, and responding with the appropriate level of intensity to the available resilience.

The net impact is a change from a dashboard-based monitoring system towards an evidence-based operational decision-making process that is more in tandem with the complexity of a large private cloud. Another outcome was related to the uniformity of operator-facing explanations. When a warning was called, the framework would typically preserve the same causal narrative as more telemetry came in, except when there was subsequently new material information about the possible source. This helped minimize churn when it comes to diagnostics, as alert after alert indicates something unrelated in the process, making the operator restart the investigation. The context layer was also used to differentiate between transients related to deployment and deterioration in the infrastructure based on the recent change activity as well as the runtime behaviour. Because of this, release windows might be monitored closely, but not necessarily considered failures. Such outcomes as the separation of advisory and actionable risk were another beneficial result. There were some observations that were deemed to be worthy of attention but not yet ready to be automated responses due to resilience being strong or due to the uncertainty regarding the suspected cause. Others were quickly promoted due to numerous signals concurring and the service impacted had very poor tolerance. This progressive kind of treatment rendered the structure more realistic as compared to a dual alarm system. The analysis also revealed that there was an effective audit trail in the decision loop. Every warning might have its evidence behind it, every suggestion to the policy rule that made it possible, and every subsequent consequence of the action that was taken or not. This traceability facilitates learning after an incident and enables the strategy to be used in the environment where changes in operations need to be justified. Combined, these results suggest predictive observability is most useful when it enhances operational judgement quality, instead of the number of identified anomalies.

5. Discussions

The overall results indicate that the greatest benefit of the proposed framework is to provide a link between prediction, diagnosis, and resilience – as opposed to optimizing a single detector. The best comparison of the capabilities is in Figure 2. The AI framework achieved a score of 91.7, which is 68.4 – 23.3 = 68.4 points higher than the conventional anomaly detection score. There was the highest increase of 27.4 points in incident prediction, which increased from 61.8 to 89.2. This is especially critical since the ability to predict reliability is based on detection of degradation prior to the emergence of an outage visible to the user. Root-cause localization increased from 65.3 to 88.6, also going up by 23.3 points, suggesting that multi-source telemetry and dependency context provide a more efficient means to narrow investigation than doing so with independent alerts. The conventional condition had a strong score on this indicator, but there was a 20.3-point increase from a score of 72.1 to a score of 92.4. This is reinforced in Table 1, where readiness interpretation is also included. 94.0 is the readiness level for anomaly detection, 92.0 for incident prediction, 90.0 for root cause localization, and 95.0 for recovery readiness. The results indicate that the framework is sufficiently developed to be deployed in a supervised operation in all four functions, with the slightly lower score of 90.0 on diagnosis suggesting that supervised human confirmation of high-impact corrective actions should be retained. Figure 3 provides instance-level information, which is otherwise not apparent in the capability scores. In all 189 observations, the confidence in the prediction was between 55.6% and 92.5% and resilience between 69.1 and 96.8. Overall, the 126 stable windows were 67.2% confident with an average score of 83.3 for their resilience, while the 63 incident windows were 78.7% confident with an average score of 78.1 for resilience. This contrast helps to explain the fundamental decision logic of the framework: in general, incident-prone conditions result in greater confidence in issuing a warning with less operational tolerance.

Meanwhile, there is overlap between the groups, which is why a private cloud shouldn't be a disruptive move based on a leap of faith. In some stable windows, the strong warning evidence is due to the fact that at the beginning of the burst demand, the

demand resembles the precursor of the failure, and some incident windows are not in the highest confidence regions. A moderating signal of Resilience context is thus important. Table 2 illustrates the Service Outcomes that result from the Capabilities. The availability increases from 96.8 to 99.2 – a 2.4-point increase. While the number change might not seem as significant as the capability-score increase, availability is already at the top end of the range; thus, a 2.4-point rise still equates to a substantial decrease in the amount of service interruptions. There is a 22.1-point difference between the alert noise before and after reduction, going from 38.5 to 16.4. This is operationally relevant, as an additional alarm would lead to fatigue and not resilience. The framework, however, generates fewer low-value notifications and enhances detection and prediction of significant degradation, as the notification value is related to the context. Recovery time is reduced by 17.4 units from 42.0 to 24.6, and localization of root causes and selection of actions to take are improved. There is the most significant increase in prevented escalations (from 54.7 to 83.1), generating the greatest resilience-outcome improvement of 28.4 points. This indicates that the framework isn't simply a way to lessen the time of already active incidents, but it additionally has a higher proportion of risky situations before they bubble over into dependencies. This interpretation is backed up by stability scores. The score in service availability is 96.0, in alert noise 90.0, in recovery time 91.0, and in prevented escalations 93.0 – all of which are fairly consistent throughout the benchmark – not reliant on handfuls of extreme observations. In combination, the values help to underpin an operational model that fuses information from telemetry to provide earlier warning and selectivity; dependency-aware analysis helps to diagnose situations, resilience scoring helps to inform the urgency of the response, and governed actions help to minimise unnecessary disruption.

Thus, the framework should be considered a decision layer on top of investments in monitoring, and not as a replacement for these investments. All the evidence can be gathered by Prometheus, distributed tracing, logs, infrastructure monitors, and application dashboards, while the AI layer brings context, prioritization, prediction, and coordinated response. This is an important distinction because enterprise private clouds don't necessarily have the luxury of changing out tooling in one fell swoop. There can be a gradual deployment, starting with advisory predictions, then diagnosis and then remediation with a policy. The numerical results also show this type of staging to be valid: 95.0 for recovery readiness, and 90.0 for diagnostic readiness, meaning that operational teams can have different levels of automation for different capabilities. The overall evidence therefore shows that it is better to obtain predictive reliability by making observability a continuous reasoning process that is linked to the resilience capacity and to the operational actions that need to be taken. The results also help to elucidate the function of the benchmark of 189 instances. The gap between 126 windows of stability and 63 windows of incidents is sufficient to allow some overlap between windows of stress. Still, without full recovery and true disruption, this is where fixed thresholds are likely to fall short. The average confidence of incident windows (78.7%) is 11.5 points higher than the average confidence of stable windows (67.2%), which helps provide separation, and the average resilience of incident windows (83.3%) is 5.2 points higher than stable windows (78.1%), providing a second point of view on urgency. These two measures, therefore, must be considered in conjunction with each other, and not combined into one alarm value. An observation with a high score (confidence) but a low score (resilience) on the 69.1 to 96.8 spectrum will require more action than an equally high score but with a higher score (resilience) on the spectrum. This interpretation is supported by the decrease in alert noise (16.4) and the increase in prevented escalations (83.1), as both the framework and the policy can be used to withhold disruptive responses in situations where there is risk but decreased tolerance.

6. Conclusion

This study created an AI-powered observability framework to transform information from a diverse set of private-cloud sources into actionable, forward-looking advice on reliability and action to enhance resilience. The controlled benchmark consisted of 189 instances covering steady, mixed and burst workload conditions to enable the framework to be investigated under steady and incident-prone behaviours. In the capability comparison, clear improvements were seen over the conventional monitoring capabilities, namely, anomaly detection, incident prediction, root cause localization, and recovery readiness, which improved from 68.4 to 91.7, 61.8 to 89.2, 65.3 to 88.6, and 72.1 to 92.4, respectively. The results were analysed at the instance level, which revealed that the prediction confidence for the stable windows was on average 67.2% and for the incident windows was 78.7%. The remaining operational tolerance was on average 83.3% for stable and 78.1% for incident windows, respectively, demonstrating the value of interpreting the warning strength in relation to remaining operational tolerance. There were also improvements in the outcomes of services. Availability improved from 96.8 to 99.2, alert noise reduced from 38.5 to 16.4, recovery time decreased from 42.0 to 24.6, and prevented escalations went up from 54.7 to 83.1. The results show that this framework can enhance reliability with no extra volume of alerts.

The architectural contribution is a closed loop linking the monitoring, contextual analysis, prediction, diagnosis, resilience assessment, governed response and feedback. This type of loop enables private-cloud operators to get in the way sooner, and still maintain auditability and human oversight. Thus, the framework could be a layer of intelligence over current observability tools, especially in scenarios where continuity of service, controlled operation and change, and explainability are paramount. This means that organisations don't have to abandon investments in monitoring in order to reap these benefits. They can implement it in steps as an advisory layer, test the predictions against existing processes for handling incidents, and then switch

on bounded automation for those incidents with well-defined paths for rollback. The readiness scores in the range from 90.0 to 95.0 justify this incremental approach, as different functions have different automation levels, depending on the degree of trust. Future production validation should focus on long-duration live telemetry, the integration of the platform, the human approval behaviour and resilience under rare cases of cascading faults. Despite the constraints of a synthetic benchmark done in a controlled environment, the results found almost always point to a model of operations where prediction is combined with context, diagnosis, awareness of resilience and action that is governed.

6.1. Limitations

There are a number of limitations to the study which should inform the interpretation of the results presented. While synthetic, the 189-instance benchmark was created with the following fields and anomaly patterns that mirror publicly documented microservice observability research. As such, it cannot simulate all the dependencies, workload shifts, configuration mistakes, hardware issues and organizational restrictions that exist in a production private cloud. The framework was tested as a single deployment, and not as part of a named enterprise platform, which means that there is no measurement of effort required to integrate the framework with virtualization stacks, storage solutions, service meshes, ticketing and change-management systems. Also, the benchmark is indicative of a range of acceptable service behaviour. The quality of predictions can be impacted by long-running seasonal effects, infrequent cascading failures, silent data corruption, partial network partitions and multi-tenant interference in a complex environment. Controlled policy logic was used to evaluate action recommendations, rather than autonomous remediation, which would enhance the safety aspect but would be beyond the scope of the current paper to consider the effect of complete autonomous remediation. Limitations are human factors, too. A number of factors, such as operator trust, alert interpretation, delay for approval and team expertise, are potential factors that could materially impact whether the predictive guidance helps improve the real incident outcomes. Further, the quality of the topology, capacity, resilience and recovery information that are provided to the framework all contribute to resilience scoring. Recommendations might be reduced in the event of incomplete service maps or stale configuration data. Last but not least, the study focuses on operational security and does not contain a specific security assessment for Telemetry Integrity, Model Manipulation or Adversarial Behaviour. These constraints are to consider the results as a proof of the architectural feasibility and controlled performance, and not as a promise of the same results in every private-cloud deployment.

6.2. Future Scope

In the future, validation of the framework, using long-duration telemetry from existing operational private clouds and multiple underlying platform architectures, is planned. The first extension is to consume the native OpenTelemetry traces, Prometheus metrics, structured logs, change events and service-topology updates from live systems in order to measure the behavior of the models in a real release cycle and/or workload seasonality. Online adaptation (adding data to the prediction layer without disruptive retraining) can add data from confirmed incidents to the prediction layer. One of the alternative directions is increased dependency reasoning. Dynamic service graphs, storage relationships, network paths and shared-resource relationships can be correlated, and the framework can determine not only a potentially faulty service, but the propagation path through which degradation can flow. Real spare capacity and replica placement, recovery-point objectives, and success history of rollbacks can also be included to make the resilience assessment more precise. Another vital extension is the automation of policies in a safe way. Future deployments should consider progressive deployment with different action levels starting with recommendations, and then automatic low-risk remediation further down the line with enough evidence being gathered. Human feedback should be made an explicit learning signal, and the operator's acceptance/rejection of the recommendations and modification of the recommendations should be used to improve subsequent decisions. Additionally, security assessment is required, as observability pipelines can be impacted by a lack of telemetry, malicious events, corrupted agents, or operational data. Lastly, larger benchmarking can be done to compare the framework for various sizes of private clouds, types of workloads and families of failures. These studies would provide generalisability, insights into platform-specific challenges, and guidance for organizations looking at a phased approach to implementing predictive observability and resilience automation.

Acknowledgment: The author sincerely acknowledges Motivity Labs Inc. for its support and contributions, which facilitated the successful completion of this research.

Data Availability Statement: The data supporting this study are available from the corresponding author upon reasonable request.

Funding Statement: The author received no external funding or financial assistance for this research.

Conflicts of Interest Statement: The author declares no conflicts of interest, and all sources used have been appropriately acknowledged.

Ethics and Consent Statement: The author ensured ethical compliance, obtained informed consent from participants, and maintained confidentiality throughout the study.

References

1. M. Farshchi, J. G. Schneider, I. Weber, and J. Grundy, "Experience report: Anomaly detection of cloud application operations using log and cloud metric correlation analysis," In *Proceedings of the IEEE 26th International Symposium on Software Reliability Engineering (ISSRE)*, United States of America, 2015.
2. A. Balalaie, A. Heydarnoori, and P. Jamshidi, "Microservices architecture enables DevOps: Migration to a cloud-native architecture," *IEEE Software*, vol. 33, no. 3, pp. 42–52, 2016.
3. L. Bao, Q. Li, P. Lu, J. Lu, T. Ruan, and K. Zhang, "Execution anomaly detection in large-scale systems through console log analysis," *Journal of Systems and Software*, vol. 143, no. 9, pp. 172–186, 2018.
4. M. Landauer, F. Skopik, M. Wurzenberger, and A. Rauber, "System log clustering approaches for cyber security applications: A survey," *Computers & Security*, vol. 92, no. 5, pp. 1–17, 2020.
5. L. Meng, F. Ji, Y. Sun, and T. Wang, "Detecting anomalies in microservices with execution trace comparison," *Future Generation Computer Systems*, vol. 116, no. 3, pp. 291–301, 2021.
6. B. Steenwinckel, D. De Paepe, S. Vanden Hautte, P. Heyvaert, M. Bentefrit, P. Moens, A. Dimou, B. Van Den Bossche, F. De Turck, S. Van Hoecke, and F. Ongenae, "FLAGS: A methodology for adaptive anomaly detection and root cause analysis on sensor data streams by fusing expert knowledge with machine learning," *Future Generation Computer Systems*, vol. 116, no. 3, pp. 30–48, 2021.
7. D. A. Bhanage, A. V. Pawar, and K. Kotecha, "IT infrastructure anomaly detection and failure handling: A systematic literature review focusing on datasets, log preprocessing, machine and deep learning approaches and automated tool," *IEEE Access*, vol. 9, no. 11, pp. 156392–156421, 2021.
8. P. Notaro, J. Cardoso, and M. Gerndt, "A survey of AIOps methods for failure management," *ACM Transactions on Intelligent Systems and Technology*, vol. 12, no. 6, pp. 1–45, 2021.
9. M. Catillo, A. Pecchia, and U. Villano, "AutoLog: Anomaly detection by deep autoencoding of system logs," *Expert Systems with Applications*, vol. 191, no. 4, pp. 1–60, 2022.
10. M. Usman, S. Ferlin, A. Brunstrom, and J. Taheri, "A survey on observability of distributed edge and container-based microservices," *IEEE Access*, vol. 10, no. 7, pp. 86904–86919, 2022.
11. M. Khanahmadi, A. Shameli-Sendi, M. Jabbarifar, Q. Fournier, and M. Dagenais, "Detection of microservice-based software anomalies based on OpenTracing in cloud," *Software: Practice and Experience*, vol. 53, no. 8, pp. 1681–1699, 2023.
12. J. Soldani and A. Brogi, "Anomaly detection and failure root cause analysis in (micro) service-based cloud applications: A survey," *ACM Computing Surveys*, vol. 55, no. 3, pp. 1–39, 2023.
13. R. Xin, P. Chen, and Z. Zhao, "CausalRCA: Causal inference based precise fine-grained root cause localization for microservice applications," *Journal of Systems and Software*, vol. 203, no. 9, pp. 1–13, 2023.
14. M. Landauer, S. Onder, F. Skopik, and M. Wurzenberger, "Deep learning for anomaly detection in log data: A survey," *Machine Learning with Applications,* vol. 12, no. 6, pp. 1–19, 2023.
15. J. Diaz-de-Arcaya, A. I. Torre-Bastida, G. Zarate, R. Minon, and A. Almeida, "A joint study of the challenges, opportunities, and roadmap of MLOps and AIOps: A systematic survey," *ACM Computing Surveys,* vol. 56, no. 4, pp. 1–30, 2023.
16. M. Panahandeh, A. Hamou-Lhadj, M. Hamdaqa, and J. Miller, "ServiceAnomaly: An anomaly detection approach in microservices using distributed traces and profiling metrics," *Journal of Systems and Software*, vol. 209, no. 3, pp. 1–23, 2024.
17. C. Almodovar, F. Sabrina, S. Karimi, and S. A. Azad, "LogFiT: Log anomaly detection using fine-tuned language models," *IEEE Transactions on Network and Service Management*, vol. 21, no. 2, pp. 1715–1723, 2024.
18. Y. Han, Q. Du, Y. Huang, P. Li, X. Shi, J. Wu, P. Fang, F. Tian, and C. He, "Holistic root cause analysis for failures in cloud-native systems through observability data," *IEEE Transactions on Services Computing,* vol. 17, no. 6, pp. 3789–3802, 2024.
19. Y. Song, R. Xin, P. Chen, R. Zhang, J. Chen, and Z. Zhao, "Autonomous selection of the fault classification models for diagnosing microservice applications," *Future Generation Computer Systems*, vol. 153, no. 4, pp. 326–339, 2024.
20. X. Ma, Y. Li, J. Keung, X. Yu, H. Zou, Z. Yang, F. Sarro, and E. T. Barr, "Practitioners' expectations on log anomaly detection," *IEEE Transactions on Software Engineering*, vol. 51, no. 9, pp. 2455–2471, 2025.

Publisher's Note: *The publisher remains impartial concerning jurisdictional claims in published maps and institutional affiliations. Responsibility for the content rests entirely with the authors and does not necessarily reflect the publisher's perspectives.*