Chaithanya Kotla[1], Krishna kanth Thottempudi[2], Radhika Kande[3], Pradeep Anjuru[4], Sivaprakash Nithyanandam[5]
[1]Devops and Cloud lead, State of Maryland, USA
[2]Infosys Limited, USA
[3]Sagarsoft Inc, USA
[4]Akkodis, USA
[5]Electrify America, USA

Received: 05-07-2024; Revised: 15-08-2024; Accepted: 28-09-2024

Self-Tuning Observability: A Meta-Learning Framework for Autonomous Monitoring Configuration and Alert Fatigue Reduction in Evolving Data Platforms

Abstract
Observability in evolving data platforms is increasingly limited by the inability of static monitoring configurations to keep pace with changing workloads, dependency structures, and signal behavior. Alert fatigue has become a major operational problem because excessive low-value notifications reduce operator attention, slow triage, and weaken trust in observability systems. The main gap is the lack of frameworks that can autonomously retune monitoring thresholds, suppression policies, and correlation logic as platform conditions change. This matters because noisy or outdated alerting policies can overwhelm operators while still failing to maintain stable monitoring quality. This article presents a meta-learning framework for self-tuning observability that adapts monitoring configuration across logs, metrics, traces, and event streams in evolving data platforms. The results show progressive stabilization of alert volume across adaptation cycles and stronger monitoring precision with greater alert fatigue reduction than static and heuristic configuration strategies. The study demonstrates that meta-learning can provide an effective foundation for autonomous observability control in dynamic data environments.
Keywords: observability, meta-learning, alert fatigue, monitoring configuration, autonomous adaptation, data platforms.

1. Introduction

Observability in modern data platforms has moved far beyond basic health dashboards and static threshold alerts. Data systems now operate across streaming engines, orchestration layers, warehouse services, feature pipelines, and governance components, which means that monitoring quality depends on how well configuration rules adapt to changing technical conditions. Alert overload has become a persistent operational problem because large volumes of logs, traces, metrics, and anomaly signals can quickly exceed the attention capacity of human operators [1]. Prioritization research has also shown that the practical challenge is not only detecting abnormal events, but deciding which ones are worthy of immediate response under noisy and fast-changing conditions [2]. A monitoring system that generates too many weak alerts can become as harmful as one that misses true incidents.

The burden is intensified when the platform itself keeps evolving. New data products, changing orchestration patterns, schema drift, infrastructure upgrades, and workload bursts all alter the baseline behavior of the system. Human-factor analysis in AI-driven operations environments has shown that repeated exposure to false positives and low-value alerts increases cognitive strain and reduces trust in automated signals [3]. Platform practitioners also report that monitoring policies often lag behind system changes because configuration tuning remains manual, fragmented, and reactive [4]. As a result, the monitoring stack may continue using thresholds, suppression rules, and correlation settings that were appropriate for an earlier platform state but no longer match the current one.

The central technical gap is that most observability systems still depend on manually maintained alert definitions and static tuning strategies. Even when machine learning is introduced, it is often used only for anomaly scoring, while the surrounding monitoring configuration remains fixed. This becomes inefficient in evolving data platforms where alert usefulness changes with workload shape, infrastructure composition, and service dependency structure. Research across intelligent operational systems suggests that adaptive learning frameworks are more effective when they optimize not just predictions, but the control policies surrounding those predictions [5]. A monitoring architecture must therefore learn how to retune itself rather than simply report abnormality scores.

This problem matters because alert fatigue directly affects response speed, issue triage quality, and the reliability of production data services. When operators repeatedly encounter noisy or weak alerts, real incidents may be delayed, dismissed, or buried inside excessive notification traffic. The consequence is not only human burden but also degraded service continuity, weaker incident accountability, and reduced confidence in observability tooling. Survey work on ML governance has similarly shown that operational trust depends on whether automated systems remain interpretable and behaviorally stable as conditions change [6]. In data platforms, that trust is closely tied to the consistency and usefulness of the alerting layer.

This study introduces a self-tuning observability framework based on meta-learning for autonomous monitoring configuration and alert fatigue reduction in evolving data platforms. The proposed design treats monitoring configuration as an adaptive policy problem in which the system observes signal behavior, evaluates alert outcomes, and updates thresholds, suppression logic, and cross-signal correlations with minimal manual intervention. The framework is intended to reduce low-value alert traffic while preserving sensitivity to meaningful incidents under platform drift and workload change. Rather than focusing only on anomaly detection, it targets the broader challenge of monitoring policy adaptation. The result is a platform-operations intelligence architecture designed to reduce operator burden while sustaining monitoring quality.

2. Methodology

The proposed framework models observability tuning as a meta-learning problem over recurring monitoring tasks drawn from evolving data platform conditions. Each task represents a different operational context, such as normal batch execution, bursty streaming ingestion, schema instability, orchestration retries, warehouse lag, or mixed incident regimes. Meta-learning literature has shown that adaptation becomes more efficient when the system learns reusable adjustment behavior across related tasks rather than re-optimizing from scratch for each new condition [7]. In this design, the monitoring controller is trained to infer which alerting parameters should change when signal distributions and incident patterns shift. The method therefore focuses on configuration adaptation rather than only abnormality detection.

The signal representation layer integrates metrics, logs, traces, and event-stream indicators into a unified monitoring state. Metrics contribute rate, latency, backlog, and utilization patterns, while logs provide error signatures and semantic change cues. Traces capture dependency paths and localized latency inflation, and event streams describe orchestration transitions, retries, and state anomalies [8]. These heterogeneous signals are encoded into a compact state vector that summarizes current platform behavior, recent alert history, operator feedback, and recent suppression outcomes. The resulting state is meant to support fast adaptation without requiring full reprocessing of raw observability data at every update cycle.

The configuration action space is defined over the parameters that most strongly influence alert behavior in real operational systems. These include threshold adjustment, severity remapping, correlation-window resizing, alert deduplication sensitivity, cooldown length, suppression scope, routing policy, and composite trigger weighting. Instead of treating these settings as fixed engineering artifacts, the framework regards them as tunable decisions whose value depends on the current platform state. Multi-objective optimization work has shown that policy tuning becomes more effective when the search

space explicitly captures tradeoffs among competing operational outcomes [9]. In the present setting, those outcomes include alert precision, incident recall, operator load, and adaptation stability.

The meta-learning controller operates through episodic adaptation cycles. During each cycle, the controller observes the current state, proposes a monitoring configuration update, and then receives feedback from subsequent alert behavior and incident outcomes. Instance-level meta-learning results in related anomaly and outlier settings have shown that task-specific adaptation can be improved when the controller learns from localized response patterns instead of relying only on global rules [10]. Following that idea, the proposed framework tracks how configuration changes affect both aggregate alert traffic and incident-specific discrimination quality. This allows the controller to learn not only whether alert volume decreased, but whether the reduction was operationally safe. The adaptation loop is therefore optimized around usable monitoring quality, not blind suppression.

Reward design is crucial because aggressive alert reduction can easily degrade true-incident visibility. The framework therefore uses a balanced reward composed of four terms: reduction in low-value alert volume, preservation of detection sensitivity, consistency of alert explanations, and stability of configuration behavior across adjacent cycles. Advisor-style meta-learning approaches in noisy environments suggest that adaptation quality improves when the learner is penalized for overreacting to unstable or misleading signals [11]. For that reason, the reward includes explicit penalties for oscillatory configuration changes, severe false-negative growth, and abrupt routing shifts that confuse operators. This makes the controller conservative toward unstable retuning patterns while still encouraging useful adaptation.

Cross-platform generalization is addressed by training over diverse operational regimes rather than over a single static platform trace. Historical workloads from multiple platform states are segmented into tasks that vary in scale, incident mix, signal sparsity, and configuration sensitivity. The controller is then meta-trained to produce good initialization behavior for unseen but related operational contexts. This is important because evolving data platforms rarely revisit the exact same condition twice. The objective is not to memorize one tuning pattern, but to learn how to tune quickly when the platform changes in familiar but non-identical ways.

Deployment is organized as a layered observability service positioned beside the existing monitoring stack. Raw signals continue to flow through the standard telemetry path, while the self-tuning layer periodically evaluates state summaries and recommends or applies bounded configuration updates. A guardrail mechanism restricts the magnitude and frequency of any one adjustment so that monitoring policy does not drift too aggressively during volatile periods. Operator override remains available, and the system records whether accepted or rejected updates later improve alert outcomes. This creates a learning loop that respects operational control while still enabling automation.

The methodology is implemented around a scenario-driven configuration model that links signal categories, tunable parameters, and adaptation targets across the platform. Monitoring tasks are replayed through successive adaptation cycles, and the controller is evaluated on its ability to stabilize alert behavior under changing platform conditions while preserving incident visibility. The overall design is meant to reduce repetitive manual retuning and make monitoring more resilient to system drift. The resulting framework supports autonomous configuration improvement without requiring a full redesign of the observability stack.

Table 1. Monitoring Signal Categories, Configuration Parameters, and Meta-Learning Adaptation Targets in Evolving Data Platforms

Signal Category	Configuration Parameters	Adaptation Target
Infrastructure metrics	threshold bands, cooldown windows	burst-aware alert stability
Pipeline execution logs	severity remapping, deduplication rules	low-value alert reduction
Distributed traces	correlation windows, dependency sensitivity	incident localization quality
Orchestration events	trigger weights, retry suppression	workflow anomaly discrimination
Data quality signals	escalation thresholds, routing rules	false positive control
Composite alert streams	policy blending, suppression scope	operator burden reduction

3. Results and Discussion

Monitoring behavior changed substantially once configuration adaptation was allowed to respond to platform drift instead of remaining fixed. The most visible effect was a progressive stabilization of alert traffic under changing system conditions, particularly when workload shifts and dependency fluctuations would normally trigger bursts of redundant notifications. This matters because alert fatigue is driven not only by absolute alert count, but by volatility in alert count across time and operational modes. A self-tuning policy is therefore useful only if it reduces both persistent alert excess and unstable alert swings. The evaluation focuses on whether the proposed meta-learning framework achieves that balance while preserving meaningful detection behavior.

Figure 1 shows how alert volume evolves across adaptation cycles under different platform conditions. The trajectories indicate that the early cycles still contain substantial noise because the controller has not yet accumulated enough task-specific response information to separate useful alerts from weak or repetitive ones. As adaptation proceeds, the volume curves flatten and begin to converge toward a more stable operating region. This is important because stabilization, rather than mere reduction, reflects that the configuration policy is becoming better aligned with the current signal environment. The downward adjustment is therefore not a simple suppression effect, but evidence of improved calibration between platform state and monitoring policy.

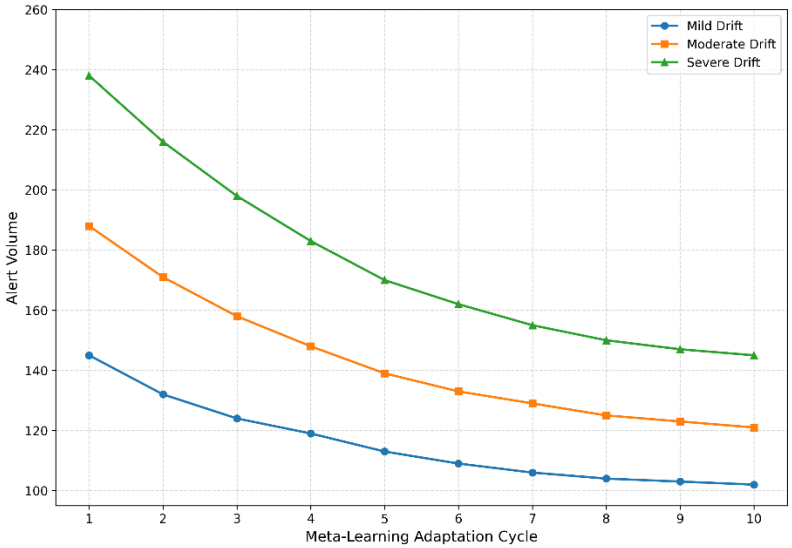

Figure 1. Alert Volume Stabilization Across Meta-Learning Adaptation Cycles Under Changing Platform Conditions

The shape of the curves also suggests that adaptation difficulty depends on the degree of platform change. Mild workload drift allows faster convergence because the signal distribution remains close to previously observed patterns. More disruptive conditions, such as concurrent orchestration changes and bursty event ingestion, require additional cycles before alert behavior becomes stable. Even in those settings, however, the controller eventually reduces redundant alert amplification without collapsing sensitivity altogether. This supports the claim that a meta-learned policy can transfer useful tuning behavior across related but non-identical platform states. The framework therefore behaves like a reusable adaptation mechanism rather than a task-specific heuristic.

Figure 2 shows monitoring precision and alert fatigue reduction are compared across autonomous configuration strategies. Static rule baselines show weaker improvement because they can only suppress alerts according to predefined logic, even when the underlying signal environment has shifted. Lightweight adaptive strategies reduce some noise, but they often do so unevenly and at the cost of inconsistent precision. The proposed meta-learning framework achieves a more favorable joint profile because it learns which configuration changes preserve incident relevance while shrinking low-value alert traffic. This is a stronger outcome than simple alert suppression because it reduces human burden without turning the monitoring layer blunt.

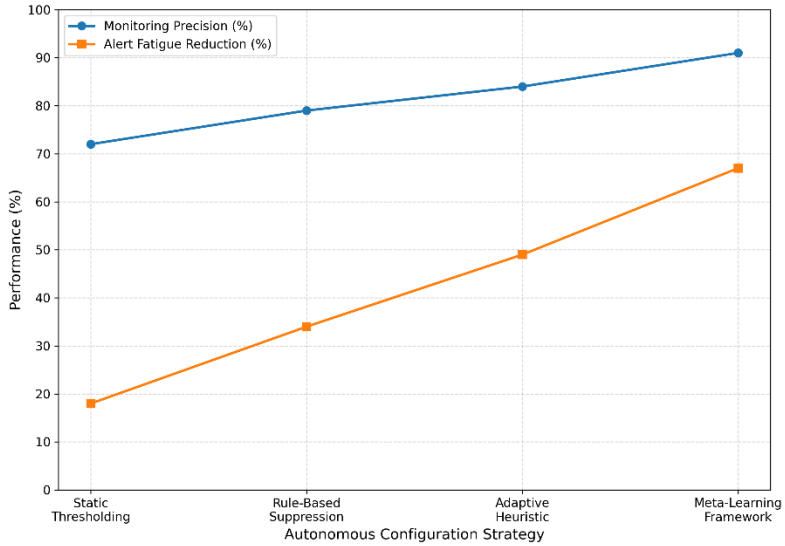

Figure 2. Monitoring Precision and Alert Fatigue Reduction Across Autonomous Configuration Strategies

That difference is especially significant for production data platforms where the cost of poor observability is both technical and organizational. A noisy monitoring policy creates repeated cognitive interruptions, while an over-suppressed policy risks missing the early signals of pipeline degradation, quality drift, or orchestration failure. The results indicate that autonomous configuration can improve this tradeoff when adaptation is tied to cross-task learning and bounded by stability-aware reward design. In effect, the system becomes better at deciding when to alert, how strongly to alert, and when not to alert at all. This makes the framework useful not only for telemetry optimization but also for reducing operational burnout in long-running data environments.

4. Conclusion

Observability becomes harder to manage as data platforms evolve, not because signals disappear, but because the meaning and usefulness of monitoring configurations change faster than manual tuning can keep up. The framework presented in this article addresses that challenge by treating alerting policy as an adaptive control problem rather than as a fixed ruleset. Through meta-learning, the system learns how to retune thresholds, suppression behavior, and correlation settings across changing operational contexts. This shifts observability from static configuration maintenance toward autonomous monitoring intelligence. The result is a more resilient monitoring layer for dynamic data environments.

The main contribution of the study lies in combining cross-signal observability, bounded configuration adaptation, and alert-fatigue-aware reward design within a single operational framework. The results indicate that the approach reduces redundant alert traffic, stabilizes monitoring behavior under platform drift, and improves the balance between alert precision and operator burden. These gains are important

because alert fatigue is not only a usability issue, but also a reliability issue for production data platforms. A monitoring stack that overwhelms operators ultimately weakens incident response itself. The proposed design therefore improves both technical monitoring quality and human-operational sustainability.

A further implication is that future data platforms may benefit from embedding learning directly into the monitoring-control layer instead of reserving intelligence for anomaly detection alone. As observability environments continue to grow in complexity, systems that can adapt their own configuration will become increasingly valuable. Future work can extend this direction toward policy transfer across organizations, human-in-the-loop reinforcement for sensitive alert classes, and deeper integration with incident management workflows. These extensions would further strengthen the practical value of self-tuning observability. The present framework offers a foundation for that transition.

References

1. Baruwal Chhetri, M., Tariq, S., Singh, R., Jalalvand, F., Paris, C., & Nepal, S. (2024). Towards human-AI teaming to mitigate alert fatigue in security operations centres. *ACM Transactions on Internet Technology*, *24*(3), 1-22.
2. Jalalvand, F., Baruwal Chhetri, M., Nepal, S., & Paris, C. (2024). Alert prioritisation in security operations centres: A systematic survey on criteria and methods. *ACM Computing Surveys*, *57*(2), 1-36.
3. Hagen, R. A., Øverlier, L., & Helkala, K. (2025). Human factors in AI-driven cybersecurity: Cognitive biases and trust issues. *Digital Threats: Research and Practice*, *6*(4), 1-20.
4. Naveed, H., Grundy, J., Arora, C., Khalajzadeh, H., & Haggag, O. (2025, September). Understanding Practitioners' Perspectives on Monitoring Machine Learning Systems. In *2025 IEEE International Conference on Software Maintenance and Evolution (ICSME)* (pp. 743-754). IEEE.
5. Dritsas, E., & Trigka, M. (2025). Machine learning in e-commerce: Trends, applications, and future challenges. *IEEE Access*.
6. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., & Galstyan, A. (2021). A survey on bias and fairness in machine learning. *ACM computing surveys (CSUR)*, *54*(6), 1-35.
7. Gharoun, H., Momenifar, F., Chen, F., & Gandomi, A. H. (2024). Meta-learning approaches for few-shot learning: A survey of recent advances. *ACM Computing Surveys*, *56*(12), 1-41.
8. Yuan, J., Xu, M., Zhao, Y., Bian, K., Huang, G., Liu, X., & Wang, S. (2022). Resource-aware federated neural architecture search over heterogeneous mobile devices. *IEEE Transactions on Big Data*.

9. Benmeziane, H., Ouarnoughi, H., El Maghraoui, K., & Niar, S. (2023). Multi-objective hardware-aware neural architecture search with Pareto rank-preserving surrogate models. *ACM Transactions on Architecture and Code Optimization*, *20*(2), 1-21.
10. Vu, L., Kirchner, P., Aggarwal, C. C., & Samulowitz, H. (2024, August). Instance-Level Metalearning for Outlier Detection. In *IJCAI* (pp. 2379-2387).
11. Ricci, S., Uricchio, T., & Del Bimbo, A. (2023). Meta-learning advisor networks for long-tail and noisy labels in social image classification. *ACM Transactions on Multimedia Computing, Communications and Applications*, *19*(5s), 1-23.