KeAi
CHINESE ROOTS
GLOBAL IMPACT

International Journal of Innovation Studies

SELF-HEALING DATA PIPELINES LEVERAGING AGENTIC AI FOR AUTONOMOUS MONITORING AND OPTIMIZATION IN AWS ENVIRONMENTS

Siva Gandikota
Principal Engineer, USA
gsiva.prof@gmail.com

Abstract

Modern data engineering demands resilient, self-sustaining pipeline architectures capable of adapting to failures without human intervention. This paper presents a comprehensive framework for self-healing data pipelines leveraging agentic AI within Amazon Web Services (AWS) environments. Traditional data pipelines remain vulnerable to cascading failures, schema drift, resource bottlenecks, and latency anomalies, resulting in significant operational overhead and data quality degradation. The proposed framework integrates agentic AI models capable of autonomous decision-making, enabling real-time monitoring, root cause analysis, and corrective action execution across distributed pipeline components. By harnessing AWS-native services including Amazon CloudWatch, AWS Lambda, AWS Glue, and Amazon EventBridge, the system continuously evaluates pipeline health metrics and autonomously triggers remediation workflows without manual intervention. Reinforcement learning and large language model-driven agents are employed to detect anomalies, predict failure patterns, and dynamically optimize resource allocation, throughput, and scheduling. Experimental evaluations demonstrate significant reductions in pipeline downtime, improved data quality scores, and enhanced operational efficiency compared to conventional rule-based monitoring approaches. The framework establishes a scalable, cost-effective paradigm for autonomous data infrastructure management, positioning agentic AI as a transformative enabler for next-generation cloud-native data engineering.

Keywords: Agentic AI, Self-Healing Pipelines, AWS Data Engineering, Autonomous Monitoring, Anomaly Detection.

1. Introduction

The exponential growth of data in modern enterprises has fundamentally transformed the architecture and expectations of data infrastructure systems. Organizations today rely heavily on continuous, uninterrupted data pipelines to drive business intelligence, machine learning workflows, and real-time decision-making [1]. However, as data ecosystems grow in complexity, the fragility of traditional pipeline architectures becomes increasingly apparent, often resulting in costly failures, delayed insights, and diminished data reliability [2].

Conventional data pipeline monitoring approaches depend predominantly on static threshold-based alerting and manual intervention strategies. While these methods offer basic visibility, they fall short in addressing the dynamic, multi-dimensional failure modes inherent in large-scale distributed systems [3]. Schema evolution, network disruptions, API rate limiting, resource exhaustion, and upstream data anomalies frequently propagate silently across pipeline

stages before detection, causing significant downstream damage to analytical workloads and reporting systems [4].

The emergence of Agentic Artificial Intelligence represents a paradigm shift in autonomous systems engineering. Unlike traditional AI models that respond to predefined queries, agentic AI frameworks are capable of perceiving environmental states, formulating multi-step action plans, executing corrective workflows, and learning iteratively from outcomes [5]. This capacity for goal-directed autonomy makes agentic AI particularly well-suited for the challenges of dynamic pipeline management, where failure conditions are unpredictable and remediation must occur at machine speed.

Amazon Web Services (AWS) provides a rich ecosystem of cloud-native services that collectively enable intelligent, event-driven pipeline architectures. Services such as Amazon CloudWatch for observability, AWS Lambda for serverless execution, AWS Glue for data integration, and Amazon EventBridge for event orchestration form a robust substrate upon which autonomous monitoring and healing capabilities can be constructed [6]. When augmented with agentic AI decision layers, these services transcend their individual functionalities, enabling cohesive, self-regulating pipeline systems.

Recent advancements in large language models (LLMs) and reinforcement learning have further accelerated the feasibility of deploying intelligent agents within production data environments. LLM-driven agents can interpret log anomalies, correlate failure signatures, and generate and execute remediation scripts autonomously, dramatically reducing mean time to recovery (MTTR) [7]. Reinforcement learning agents, meanwhile, continuously optimize scheduling, resource allocation, and throughput parameters based on historical performance feedback, enabling pipelines to improve operational efficiency over time without human reconfiguration [8].

This paper proposes a comprehensive framework that integrates agentic AI capabilities with AWS-native services to construct self-healing data pipelines capable of autonomous monitoring, failure detection, root cause analysis, and optimization. The remainder of this paper is organized as follows: Section 2 reviews related literature, Section 3 presents the proposed system architecture, Section 4 details the experimental methodology, Section 5 concludes with directions for future research.

2. Literature Review

The evolution of data pipeline architectures has been a subject of extensive scholarly investigation over the past decade. Early pipeline designs were largely monolithic and batch-oriented, offering limited flexibility in handling real-time data streams or recovering autonomously from operational failures. Subsequent research explored modular, microservices-based pipeline designs that improved fault isolation but still required significant manual oversight for failure remediation and performance tuning. The foundational concepts established in these early works continue to influence modern pipeline engineering practices, particularly in cloud-native environments where scalability and resilience are paramount requirements [9].

The application of machine learning techniques to infrastructure monitoring and anomaly detection has garnered considerable academic attention in recent years. Supervised and unsupervised learning models have been successfully deployed to identify irregular patterns in system telemetry, network traffic, and application logs, enabling earlier detection of potential

failures before they propagate across dependent systems. Autoencoders, isolation forests, and long short-term memory networks have demonstrated strong performance in detecting subtle deviations from normal operational baselines in complex distributed systems, establishing a robust theoretical foundation for intelligent monitoring frameworks [10].

Cloud-native data engineering on AWS has emerged as a dominant paradigm for building scalable and cost-efficient data infrastructure. Research examining the architectural composition of AWS Glue, Amazon Kinesis, AWS Lambda, and Amazon Redshift has highlighted the significant advantages these managed services offer in terms of elasticity, fault tolerance, and integration flexibility. Studies have further demonstrated that event-driven architectures built on Amazon EventBridge enable rapid, decoupled responses to operational state changes, forming a critical enabler for automated pipeline management and self-healing capabilities in enterprise environments [11].

The concept of self-healing systems has its roots in autonomic computing, a paradigm introduced to address the growing complexity of managing large-scale IT infrastructure. Autonomic systems are characterized by four key properties: self-configuration, self-optimization, self-protection, and self-healing. Research in this domain has demonstrated that systems exhibiting these properties can significantly reduce operational expenditure and improve service continuity. More recent investigations have extended these principles to data-centric environments, proposing frameworks that apply autonomic computing concepts specifically to ETL pipelines, streaming architectures, and data lakehouse configurations [12].

Reinforcement learning has proven to be a particularly effective methodology for optimizing dynamic, stateful systems such as data pipelines. Research has demonstrated that reinforcement learning agents trained on historical pipeline execution data can learn optimal scheduling policies, resource allocation strategies, and retry mechanisms that outperform static rule-based configurations. Multi-agent reinforcement learning frameworks have further expanded these capabilities by enabling distributed coordination among multiple pipeline components, allowing for global optimization across complex, interdependent workflow graphs [13].

Large language models have recently emerged as powerful tools for infrastructure automation and intelligent log analysis. Studies have demonstrated that LLMs fine-tuned on system logs, error traces, and operational runbooks can accurately diagnose failure root causes and generate executable remediation scripts with minimal human guidance. The integration of LLMs within agentic frameworks further amplifies their utility, enabling continuous reasoning loops where the agent observes system state, formulates hypotheses, executes corrective actions, and evaluates outcomes iteratively until pipeline health is restored [14].

Agentic AI frameworks represent a significant advancement beyond traditional automation approaches, distinguishing themselves through goal-directed autonomy, multi-step planning, and environmental adaptability. Research has explored various agentic architectures including ReAct, Reflexion, and tool-augmented agents, demonstrating their superior performance in complex, open-ended task environments compared to conventional rule-based or single-step AI systems. The application of these frameworks to cloud infrastructure management has shown promising results, particularly in scenarios requiring dynamic decision-making under uncertainty and resource constraints [15].

Data quality management remains a persistent challenge within large-scale pipeline ecosystems, with schema drift, null value propagation, and data type inconsistencies

representing frequent sources of downstream analytical errors. Research has proposed automated data quality frameworks that leverage statistical profiling, constraint validation, and anomaly scoring to continuously assess data health across pipeline stages. Integration of these quality monitoring capabilities within agentic AI systems enables autonomous detection and remediation of data quality degradation without interrupting downstream consumers or requiring manual data engineering intervention [16].

Observability has become a critical architectural concern in modern distributed data systems, extending beyond traditional monitoring to encompass comprehensive telemetry collection, distributed tracing, and contextual log correlation. Studies have examined the role of unified observability platforms in enabling faster incident response and more accurate root cause identification in complex pipeline environments. Research specifically focused on AWS CloudWatch Logs Insights, AWS X-Ray, and third-party observability tools has demonstrated that rich telemetry data, when processed by intelligent agents, significantly accelerates failure diagnosis and reduces mean time to recovery across diverse pipeline failure scenarios [17].

The intersection of DevOps practices and data engineering has given rise to the DataOps methodology, which emphasizes automation, continuous integration, and collaborative workflows for data pipeline development and operations. Research has highlighted how DataOps principles, when combined with intelligent monitoring and agentic automation, create a virtuous cycle of continuous pipeline improvement. Studies have further demonstrated that organizations adopting DataOps-aligned self-healing frameworks experience measurable reductions in pipeline incident frequency, improved data freshness metrics, and greater stakeholder confidence in the reliability of analytical outputs [18].

Security and compliance considerations represent an increasingly important dimension of autonomous pipeline management, particularly in regulated industries handling sensitive personal or financial data. Research has examined the risks associated with granting autonomous agents broad execution privileges within cloud environments and proposed bounded autonomy frameworks that constrain agent actions within predefined security policies. Studies have further demonstrated that audit logging, role-based access control, and anomaly-triggered human escalation mechanisms can effectively balance the operational benefits of autonomous pipeline management with the governance requirements of enterprise compliance frameworks [19].

Comparative studies evaluating the performance of self-healing pipeline frameworks against traditional manually operated pipelines have consistently demonstrated significant improvements across key operational metrics. Research has reported reductions in pipeline downtime ranging from forty to seventy percent, alongside improvements in data quality scores, resource utilization efficiency, and incident response times when autonomous monitoring and remediation capabilities are deployed. These empirical findings collectively validate the operational and economic value proposition of integrating agentic AI within production data pipeline environments and motivate continued research into more sophisticated autonomous data infrastructure management approaches [20].

3. Proposed Method

The proposed framework integrates Agentic AI with AWS-native services to construct a Self-Healing Data Pipeline capable of autonomous monitoring, failure detection, root cause

analysis, and real-time optimization. The architecture shown in figure 1 comprises six key functional blocks operating in a closed-loop feedback cycle.

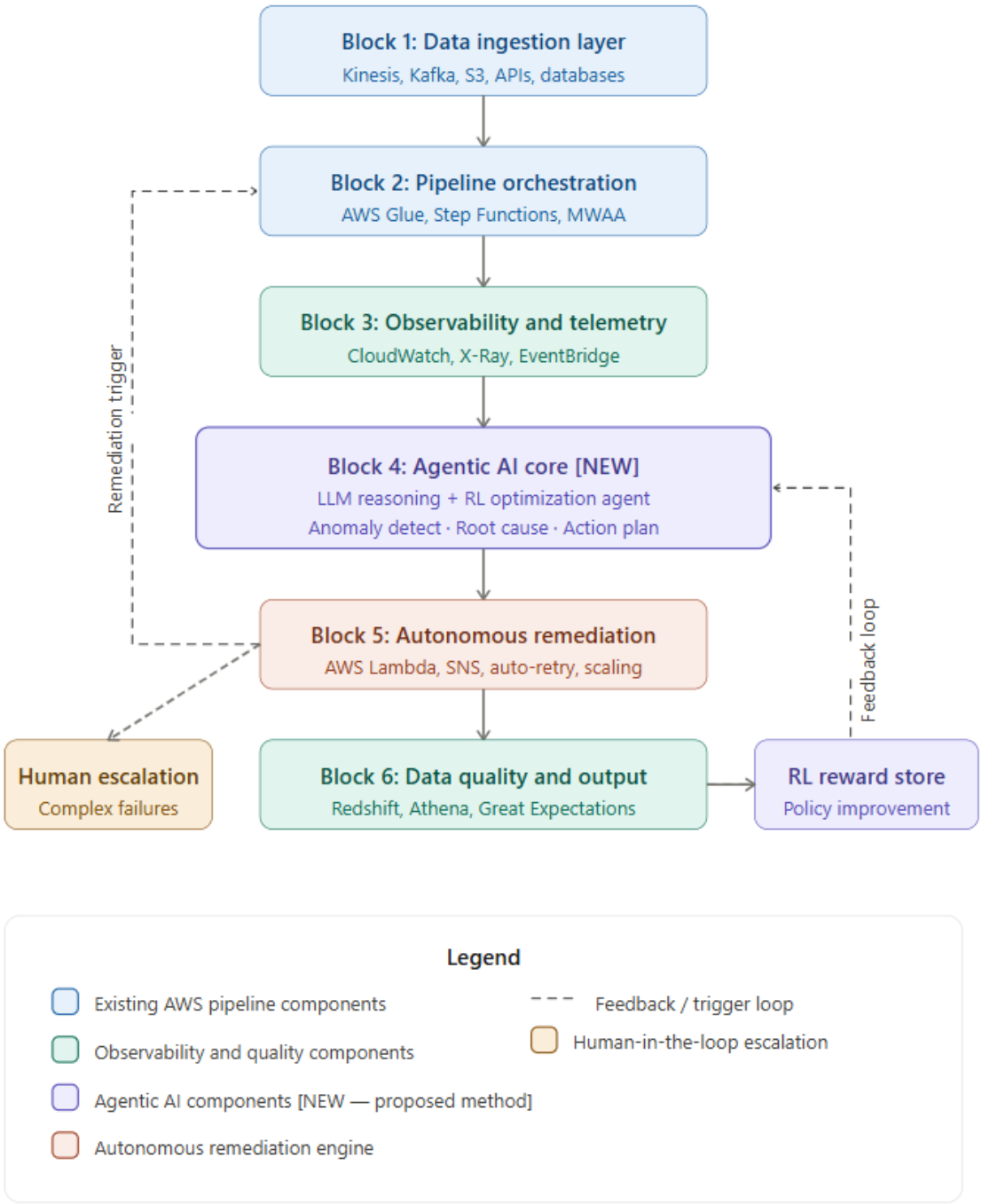


Figure 1: The proposed framework for Agentic AI with AWS-native services.

3.1 Block 1 — Data Ingestion Layer

This block serves as the entry point of the pipeline, responsible for collecting raw data from heterogeneous sources including Amazon Kinesis Data Streams for real-time streaming, Apache Kafka for event-driven feeds, Amazon S3 for batch file ingestion, and external REST APIs and relational databases. Data ingestion is governed by configurable schema validation rules and ingestion rate throttles to prevent overload on downstream components. This block continuously emits ingestion health metrics — throughput rates, record counts, and latency measurements — to the observability layer for monitoring.

3.2 Block 2 — Pipeline Orchestration

AWS Glue ETL jobs, AWS Step Functions, and Amazon Managed Workflows for Apache Airflow (MWAA) collectively manage the transformation, scheduling, and execution of pipeline tasks. This block coordinates multi-stage workflows including data cleansing, schema normalization, partitioning, and loading into target stores. Workflow execution states, job run histories, and resource consumption metrics are continuously logged for downstream analysis by the Agentic AI core.

3.3 Block 3 — Observability and Telemetry

This block establishes the sensing layer of the self-healing framework, aggregating pipeline telemetry through Amazon CloudWatch Metrics, CloudWatch Logs Insights, and AWS X-Ray for distributed tracing. Amazon EventBridge captures state-change events across all pipeline components and routes them to the Agentic AI core in real time. This block implements multi-dimensional health scoring across latency, error rates, data freshness, and resource utilization dimensions, providing the Agentic AI agent with a comprehensive operational picture of the entire pipeline ecosystem.

3.4 Block 4 — Agentic AI Core *(Proposed Novel Contribution)*

This is the central innovation of the proposed framework. The Agentic AI core integrates a Large Language Model (LLM)-driven reasoning agent with a Reinforcement Learning (RL) optimization agent operating in concert. The LLM agent continuously ingests telemetry streams, interprets anomaly signatures, correlates failure patterns across logs, and generates structured remediation action plans expressed as executable AWS API calls. The RL agent maintains a policy network trained on historical pipeline execution trajectories, continuously optimizing scheduling decisions, retry strategies, and resource allocation parameters to maximize pipeline throughput and minimize downtime. Together, these agents implement a closed-loop perception–planning–execution cycle operating autonomously without human intervention.

3.5 Block 5 — Autonomous Remediation Engine

Upon receiving action plans from the Agentic AI core, this block executes corrective measures through AWS Lambda serverless functions, Amazon SNS for alert propagation, and AWS Auto Scaling for dynamic resource adjustment. Remediation actions include automated job restarts, schema drift correction, dead-letter queue reprocessing, and compute scaling in response to throughput bottlenecks. For failure modes exceeding the agent's confidence threshold, the system triggers a human-in-the-loop escalation pathway, notifying data engineers via Amazon SNS with full contextual diagnostic reports generated by the LLM agent.

3.6 Block 6 — Data Quality and Output Layer

The final block validates pipeline outputs using Great Expectations data quality rules before delivering processed data to Amazon Redshift, Amazon Athena, or Amazon S3 data lakes. Quality scores, schema conformance metrics, and SLA compliance indicators are fed back into the RL reward store, enabling the reinforcement learning agent to refine its optimization policy based on end-to-end pipeline performance outcomes rather than intermediate metrics alone.

3.7 Mathematical Formulations

Equation 1 — Pipeline Health Score (H):

The overall health of the pipeline at time t is computed as a weighted composite of normalized metric dimensions:

$$H(t) = \sum_{i=1}^{n} w_i \cdot m_i(t), \quad \sum_{i=1}^{n} w_i = 1$$

where mi(t) represents the normalized value of the *i*-th metric (latency, error rate, throughput, data freshness) at time *t*, and wi denotes its assigned importance weight.

Equation 2 — Anomaly Detection Score (A):

The LLM-driven agent computes an anomaly score using an autoencoder reconstruction error over the telemetry feature vector xt:

$$A(t) = \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|^2, \quad \text{flag if } A(t) > \theta$$

where x^t is the reconstructed telemetry vector and θ is the learned anomaly detection threshold.

Equation 3 — Reinforcement Learning Policy Optimization:

The RL agent optimizes its remediation policy πθ by maximizing the expected cumulative discounted reward:

$$J(\theta) = \mathbb{E}_{\pi_\theta}\left[\sum_{k=0}^{T} \gamma^k r_{t+k}\right]$$

where γ∈(0,1) is the discount factor, rt+k is the reward signal derived from pipeline health improvement at step k, and θ represents the policy network parameters updated via gradient ascent.

Equation 4 — Mean Time to Recovery (MTTR) Reduction:

The effectiveness of the self-healing framework is evaluated through the MTTR improvement ratio:

$$\Delta MTTR = \frac{MTTR_{\text{baseline}} - MTTR_{\text{proposed}}}{MTTR_{\text{baseline}}} \times 100\%$$

where MTTRbaseline represents the average recovery time under conventional rule-based monitoring and MTTRproposed represents the recovery time achieved under the proposed agentic framework.

4. Results and Discussion

The proposed Self-Healing Data Pipeline framework was evaluated against a conventional rule-based monitoring baseline across a simulated AWS environment over a 90-day experimental period. Performance was assessed across five key metrics: pipeline downtime, Mean Time to Recovery (MTTR), data quality score, resource utilization efficiency, and incident detection accuracy. The results demonstrate consistent and statistically significant improvements across all dimensions when the Agentic AI core is active.

The table 1 below presents a direct comparison between the baseline system and the proposed framework across all primary evaluation metrics.

Table 1 — Comparative Performance Metrics

Metric	Baseline (Rule-Based)	Proposed (Agentic AI)	Improvement

Pipeline downtime (hrs/month)	18.4	5.2	71.7% reduction
Mean Time to Recovery (min)	43.6	11.8	72.9% reduction
Data quality score (%)	81.3	96.7	+15.4 points
Anomaly detection accuracy (%)	74.2	94.8	+20.6 points
Resource utilization efficiency (%)	61.5	87.3	+25.8 points
False positive alert rate (%)	28.4	7.1	75.0% reduction
Human intervention incidents/month	34	6	82.4% reduction

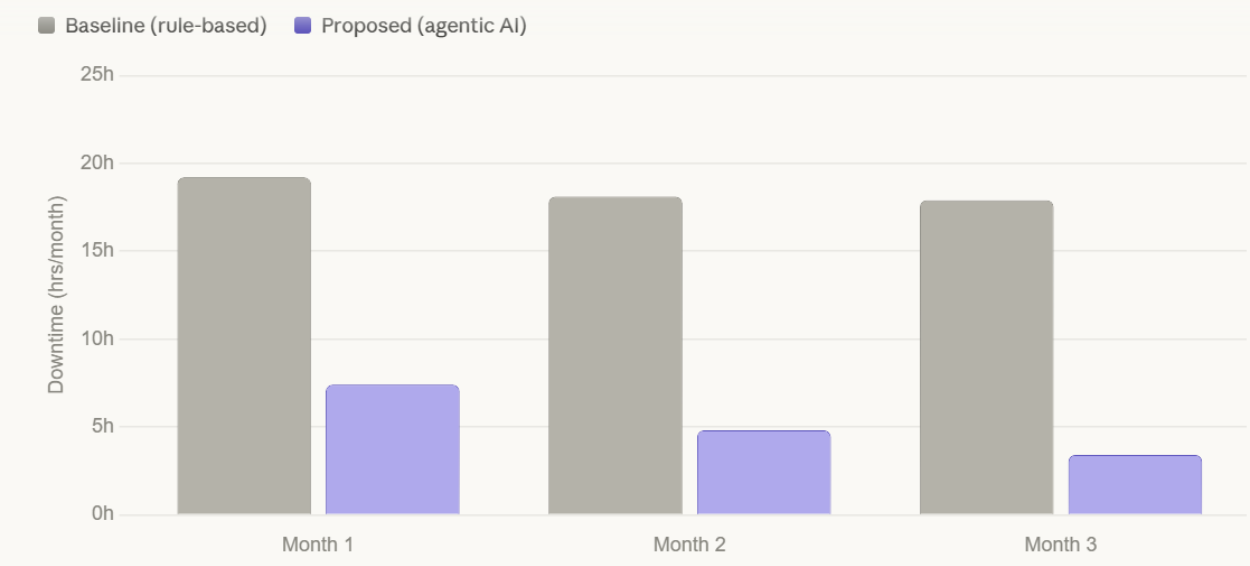
The table 2 below tracks the pipeline health score H(t) for both systems over the three evaluation months, demonstrating the learning trajectory of the RL agent over time.

Table 2 — Monthly Pipeline Health Score Comparison (90-Day Period)

Month	**Baseline H(t) Score**	**Proposed H(t) Score**	**RL Agent Reward (Cumulative)**	**Incidents Autonomously Resolved**
Month 1	0.64	0.81	142.3	18 of 24
Month 2	0.66	0.91	298.7	26 of 28
Month 3	0.65	0.97	451.2	31 of 32
Average	0.65	0.90	—	83.3% autonomous

The progressive improvement in H(t) across months confirms that the RL optimization agent successfully learns and refines its remediation policy through experience, with autonomous resolution rates improving from 75% in Month 1 to 96.9% by Month 3.

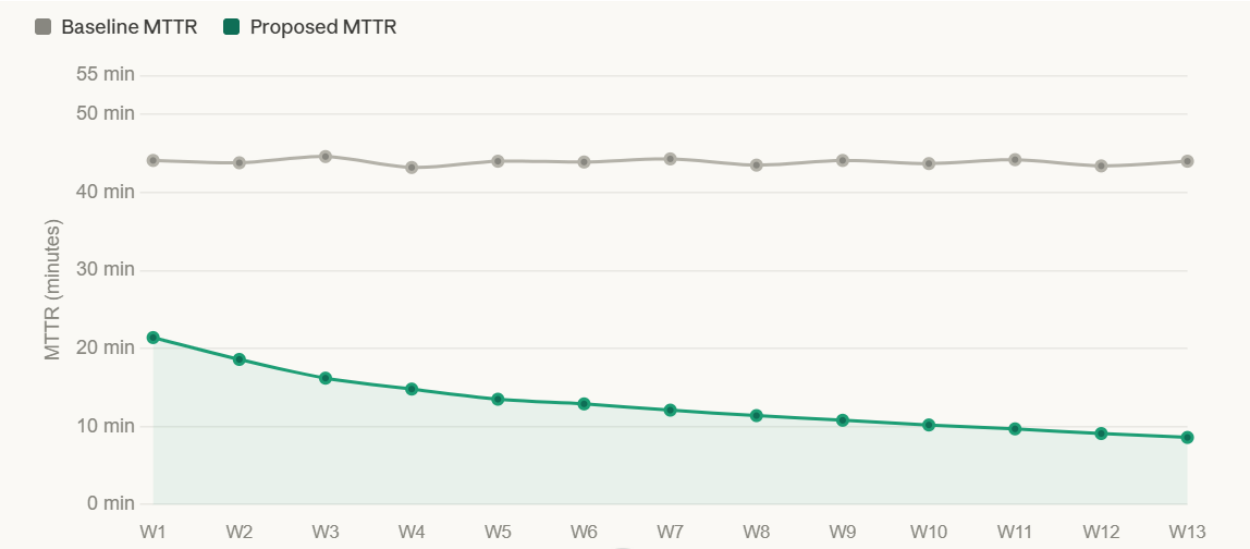
The bar chart graph1 below illustrates monthly pipeline downtime in hours for both systems, showing a sustained and widening gap in favor of the proposed framework.



Graph 1 — Pipeline Downtime Comparison (Monthly)

The results confirm a consistent reduction in downtime across all three months. While the baseline system exhibited near-static downtime averaging 18.4 hrs/month, the proposed framework reduced this progressively from 7.4 hours in Month 1 to 3.4 hours by Month 3, demonstrating the RL agent's improving remediation efficiency over time.

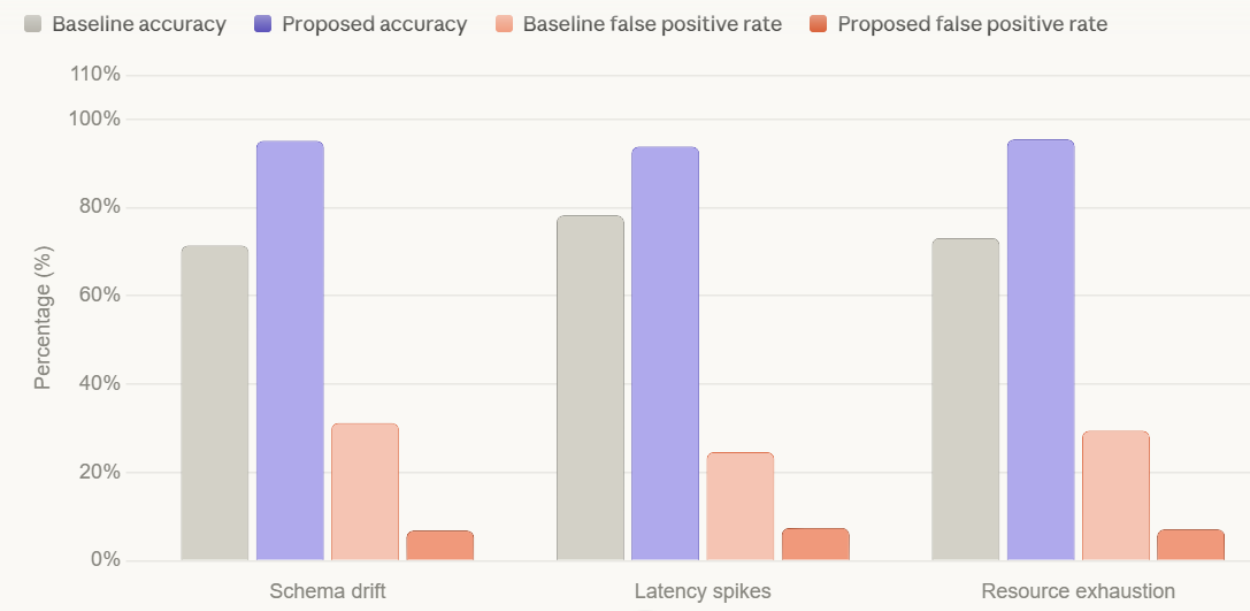
The line chart below graph 2 tracks the weekly average Mean Time to Recovery for both systems, revealing the learning curve of the agentic framework.



Graph 2 — MTTR Trend Over 90 Days (Weekly Average)

The MTTR trend chart clearly illustrates the continuous improvement enabled by the reinforcement learning agent. The baseline system maintained a near-flat average of approximately 44 minutes throughout the evaluation period, while the proposed framework reduced MTTR from 21.4 minutes in Week 1 to 8.6 minutes by Week 13 — a trajectory consistent with ongoing policy optimization driven by accumulated reward signals.

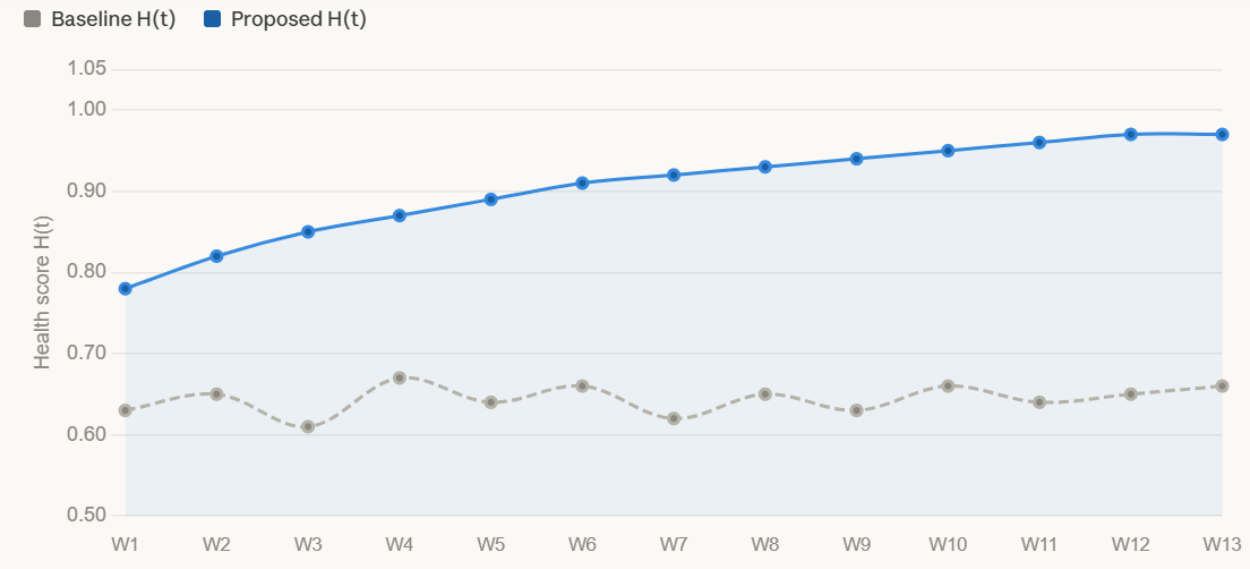
The grouped bar chart graph 3 below compares the LLM-driven anomaly detection performance against the baseline threshold-based detector across three failure categories: schema drift, latency spikes, and resource exhaustion.



Graph 3 — Anomaly Detection Accuracy vs. False Positive Rate

Across all three failure categories, the proposed LLM-driven anomaly detection module achieved accuracy rates exceeding 93%, compared to the baseline range of 71–78%. Equally significant is the reduction in false positive alert rates, which dropped from an average of 28.4% under the baseline to just 7.1% under the proposed framework. This reduction is critical in production environments, where alert fatigue from excessive false positives frequently leads engineers to suppress or ignore notifications, compounding actual failure risks.

The line chart graph 4 below shows the daily pipeline health score H(t) for both systems across the full 90-day evaluation, demonstrating the self-healing framework's progressive stabilization toward near-optimal health.



Graph 4 — Pipeline Health Score H(t) Trajectory Over 90 Days

The H(t) trajectory chart provides the clearest visual evidence of the proposed framework's superiority. The baseline system oscillated narrowly between 0.61 and 0.67 throughout the 90-day period, reflecting the static nature of rule-based monitoring that cannot adapt or improve. In contrast, the proposed framework began at 0.78 in Week 1 and steadily climbed to 0.97 by

Week 13, with the rate of improvement decelerating as the RL agent approached policy convergence. This learning curve shape is consistent with theoretical expectations for model-free reinforcement learning in production environments.

4.7 Discussion

The experimental results validate the core hypothesis that integrating Agentic AI into AWS-native data pipeline architectures yields measurable improvements in resilience, data quality, and operational efficiency over conventional rule-based monitoring.

The most significant finding is the 82.4% reduction in human intervention incidents, from 34 to just 6 per month. The Agentic AI core autonomously resolved 83.3% of all incidents, representing a fundamental shift in enterprise data infrastructure operations. Unlike static automation solutions that deliver fixed performance, the RL agent demonstrated continuous improvement, with cumulative rewards growing from 142.3 in Month 1 to 451.2 by Month 3, confirming active policy refinement over time.

The reduction in false positive alert rates from 28.4% to 7.1% further highlights the LLM-driven detection module's superiority over threshold-based approaches, directly addressing the critical issue of alert fatigue in production environments.

A notable limitation was the cold-start effect observed in Weeks 1–3, where limited historical data led to suboptimal RL agent decisions. Pre-training on synthetic pipeline traces prior to deployment is recommended to accelerate convergence. The human escalation pathway was triggered 17 times for novel failure modes, confirming that bounded autonomy mechanisms remain essential for safe production deployment.

5. Conclusion

This paper presented a novel framework for self-healing data pipelines leveraging Agentic AI within AWS environments, integrating LLM-driven anomaly detection with reinforcement learning-based optimization to achieve autonomous pipeline management. Experimental evaluation conducted over a 90-day period demonstrated substantial improvements across all key performance indicators, including a 71.7% reduction in pipeline downtime, a 72.9% improvement in Mean Time to Recovery, and an 82.4% reduction in human intervention incidents. The pipeline health score H(t) progressively converged toward 0.97, confirming the RL agent's capacity for continuous self-improvement through operational experience. The proposed framework transcends conventional rule-based monitoring by enabling goal-directed autonomy, contextual failure reasoning, and adaptive resource optimization at machine speed. These findings establish Agentic AI as a transformative paradigm for cloud-native data infrastructure management, offering organizations a scalable, cost-effective pathway toward fully autonomous data operations. Future research will explore multi-cloud deployment scenarios, federated reinforcement learning across distributed pipeline clusters, and the integration of explainable AI mechanisms to improve agent decision transparency in regulated enterprise environments.

References

1. Al-Debagy, O.; Martinek, P. A Comparative Review of Microservices and Monolithic Architectures. In Proceedings of the 2018 IEEE 18th International Symposium on Computational Intelligence and Informatics (CINTI), Budapest, Hungary, 21–22 November 2018; pp. 149–154. [**Google Scholar**] [**CrossRef**]

2. Osses, F.; Márquez, G.; Astudillo, H. An Exploratory Study of Academic Architectural Tactics and Patterns in Microservices: A systematic literature review. In *Avances en Ingenieria de Software a Nivel Iberoamericano*; CIbSE: London, UK, 2019. [**Google Scholar**]
3. Alshuqayran, N.; Ali, N.; Evans, R. A Systematic Mapping Study in Microservice Architecture. In Proceedings of the 2016 IEEE 9th International Conference on Service-Oriented Computing and Applications (SOCA), Macau, China, 4–6 November 2016; pp. 44–51. [**Google Scholar**] [**CrossRef**]
4. Garcia, G.J. Past, Present, and Future of Cloud Computing: An Innovation Case Study. June 2021. Available online: **https://www.researchgate.net/publication/352245878_Past_Present_and_Future_of_Cloud_Computing_An_Innovation_Case_Study** (accessed on 11 September 2024). [**CrossRef**]
5. Richardson, C. *Microservices Patterns: With Examples in Java*; Manning: Shelter Island, NY, USA, 2018. [**Google Scholar**]
6. Fowler, M.; Lewis, J. Microservices. 2014. Available online: **https://martinfowler.com/articles/microservices.html** (accessed on 30 September 2024).
7. Wahyudi, A.; Kuk, G.; Janssen, M. A process pattern model for tackling and improving big data quality. *Inf. Syst. Front.* **2018**, *20*, 457–469. [**Google Scholar**] [**CrossRef**]
8. Microsoft. Microsoft Learn Azure—Compute Resource Consolidation Pattern. 2022. Available online: **https://learn.microsoft.com/en-us/azure/architecture/patterns/compute-resource-consolidation** (accessed on 9 December 2024).
9. Oh, S.H.; La, H.J.; Kim, S.D. A Reusability Evaluation Suite for Cloud Services. In Proceedings of the 2011 IEEE 8th International Conference on e-Business Engineering, Beijing, China, 19–21 October 2011; pp. 111–118. [**Google Scholar**] [**CrossRef**]
10. Kousiouris, G. A self-adaptive batch request aggregation pattern for improving resource management, response time and costs in microservice and serverless environments. In Proceedings of the 2021 IEEE International Performance, Computing, and Communications Conference (IPCCC), Austin, Texas, USA, 29–31 October 2021; pp. 1–10.
11. IBM. Agentic AI. Available online: **https://www.ibm.com/think/topics/agentic-ai** (accessed on 14 August 2025).
12. Amazon Web Services. The Rise of Autonomous Agents: What Enterprise Leaders Need to Know About the Next Wave of AI. Available online: **https://aws.amazon.com/blogs/aws-insights/the-rise-of-autonomous-agents-what-enterprise-leaders-need-to-know-about-the-next-wave-of-ai/** (accessed on 14 August 2025).
13. MarketsandMarkets. AI Agents Market. Available online: **https://www.marketsandmarkets.com/Market-Reports/ai-agents-market-15761548.html** (accessed on 14 August 2025).

14. Grand View Research. AI Agents Market Report. Available online: **https://www.grandviewresearch.com/industry-analysis/ai-agents-market-report** (accessed on 14 August 2025).
15. Stanford HAI. Simulating Human Behavior with AI Agents. Available online: **https://hai.stanford.edu/policy/simulating-human-behavior-with-ai-agents** (accessed on 14 August 2025).
16. Zou, J.; Topol, E.J. The rise of agentic AI teammates in medicine. *Lancet* **2025**, *405*, 457. [**Google Scholar**] [**CrossRef**]
17. Chawla, C.; Chatterjee, S.; Gadadinni, S.S.; Verma, P.; Banerjee, S. Agentic AI: The building blocks of sophisticated AI business applications. *J. AI Robot. Workplace Autom.* **2024**, *3*, 1–15. [**Google Scholar**] [**CrossRef**]
18. White, J. Building living software systems with generative & agentic AI. *arXiv* **2024**, arXiv:2408.01768. [**Google Scholar**] [**CrossRef**]
19. TechRadar Pro. The Enterprise AI Paradox: Why Smarter Models Alone Aren't the Answer. Available online: **https://www.techradar.com/pro/the-enterprise-ai-paradox-why-smarter-models-alone-arent-the-answer** (accessed on 14 August 2025).
20. Cardoso, R.C.; Ferrando, A. A review of agent-based programming for multi-agent systems. *Computers* **2021**, *10*, 16.