International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(4s) 2026 pp. 366-374
https://doi.org/10.70917/ijcisim-2026-2522
Article

Autonomous Data Pipeline Orchestration

Yadwinder Singh Sandhu[1], Nirdesh Pachoriya[2], Aditya Rautaray[3], Bhaskar Reddy Kollu[4], Raghava Chellu[5], Sitaram Satapathy[6]*

[1] Independent Researcher.
Email: Yadwinder1944@gmail.com
[2] Fidelity Investments, USA.
Email: Nirdesh.pachoriya@ieee.org
[3] Cloud Solution Architect, AIOps & MLOps Architect, Cybersecurity Specialist, CVS Health, Corporate Headquarters, One CVS Drive, Woonsocket, Rhode Island 02895, United States.
Email: Aditya.Rautaray@cvshealth.com
[4] Enterprise Architect & Researcher, Dallas, Texas 75035, USA.
Email: bhaskar.kollu@outlook.com
ORCID: 0009-0002-9085-8301
[5] Support Engineer – Specialist, Equifax Inc., USA.
Email: raghava.chellu@gmail.com
[6] GIFT Autonomous, India.
Email: sitaram.cet@gmail.com

Abstract: Data engineering is shifting from simple ETL scripting to a more supervisory approach with human engineers setting strategic objectives, quality requirements, and compliance limits, while automated agents creating, testing, deploying, and fixing data pipelines. The intent of this research is to create and test an Autonomous Data Pipeline Orchestration framework that takes high-level pipeline requirements and transforms them into deployable workflows, adds governance checks to the pipeline generation process, and performs self-healing actions at runtime when things go wrong. The static pipeline templates have been compared with the proposed agentic framework using a controlled benchmark for 48 pipeline runs across order, transaction, sensor, and clickstream workloads. Two main parameters have been chosen to analyze: deployment-ready pipeline generation and data-quality recovery that happens automatically during deployment. The result indicates that the agentic framework deployed at 87.5% while static templates deployed at 25.0%. It also added an average of 11.5 validation controls per run versus 2.0 controls in the baseline, and the number of defects that were not resolved dropped from 5,314.9 to 416.7 per 10,000 records. The results suggest that in a modern data engineering context, the use of autonomous orchestration can help minimize manual debugging tasks, enhance governance by design, and boost reliability.
Keywords: Autonomous orchestration, data pipeline, agentic data engineering, ETL, self-healing, data governance

1. Introduction

Designed to bring together a variety of data sources, transformation engines, storage systems and downstream analytical applications into one operational environment, data pipelines have become an essential piece of the puzzle for modern analytics, artificial intelligence (AI) and enterprise decision-making systems. Data pipelines are becoming a vital tool for organizations in various industries to facilitate real-time analytics, predictive modeling, business intelligence and decision automation. The amount and complexity of data ecosystems is growing and pipelines no longer just support simple Extract-Transform-Load (ETL) processes. Rather, they run 24/7 across distributed cloud, handle streams of structured and unstructured data, and provide the reliability and governance needed by machine learning systems. Yet, despite all of these developments, today's data pipelines are extremely susceptible to operational problems and quality issues following deployment. Various problems such as schema drift, delayed event processing, missing data, type mismatches, duplicate records, inconsistent metadata, policy violations, etc., that may arise can affect downstream analytics and model reliability (Akidau et al., 2015; Munappy et al., 2020).

As a consequence of this trend toward more comprehensive analytics and AI production systems, pipeline observability, validation and governance has become more critical. Transferring data between systems isn't enough – it has to be guaranteed to be correct, converted into the correct format and monitored all the time, and also be governed by organizational policy and compliance requirements (Baylor et al., 2017; Polyzotis et al., 2018). Production machine learning environments are particularly reliant on stable and reliable pipelines as minor issues in the upstream data quality can result in incorrect predictions, model drift, or disruption in operations.

Consequently, there is growing recognition of the need for pipeline reliability to be considered as a fundamental engineering need, not an afterthought in operations.

Challenge	Description	Impact on Data Operations
Schema Drift	Unexpected changes in source data structures	Pipeline failures and deployment delays
Missing Values	Incomplete records during ingestion or transformation	Reduced analytical accuracy
Type Mismatch	Data types differ from expected formats	Processing errors and validation failures
Duplicate Records	Repeated entries in datasets	Inaccurate reporting and model bias
Governance Violations	Non-compliance with organizational policies	Regulatory and operational risks
Manual Debugging	Human intervention required for failure resolution	Increased maintenance cost and downtime

Table 1. Key Challenges in Traditional Data Pipeline Management

Current pipeline engineering methods are dependent on scripts that are written, static workflow templates, and rule-based orchestration systems. There are these cases, where engineers have to design the transformations logic, write the validation logic, track pipeline failures, and handle both the failures and manually fix pipelines when there are any. While this gives limited control, it also presents a number of practical constraints in terms of scalability, adaptability, and maintenance. Static ETL templates rarely can adapt well to dynamically changing data structures or runtime. Schemas change unexpectedly, or data-quality rules are not met, so pipelines will often need to be manually redeployed and debugged, leading to additional delays and technical debt. A study on DataOps and Pipeline Development has revealed that enterprises face the challenge of disparate tools, varying governance rules, and inadequate automation in production Data Systems (Munappy et al., 2020; Raj Munappy et al., 2020).

With the recent development of autonomous agents, large language models (LLMs), and reasoning-based software systems, there are opportunities to tackle these limitations with intelligent orchestration mechanisms. Agentic systems add a paradigm where software agents can understand goals, reason about actions, monitor runtime performance and modify their behavior within a set of constraints (Shinn et al., 2023; Wang et al., 2024; Yao et al., 2023). In the context of data engineering, this can help orchestration systems shift from rigid execution to a goal-oriented pipeline management. Autonomous agents can understand the high-level pipeline goals and translate them into transformation logic, add validation logic, track the performance of the pipeline during runtime, and start repair logic when the pipeline fails, rather than relying completely on human-written code.

However, this transformation towards agentic data engineering gives rise to the idea of Autonomous Data Pipeline Orchestration. With this method, orchestration is an ongoing and adaptive process, where agents work together to get to deployment readiness, stay compliant to governance, and ensure runtime reliability. Schema checks, null validations, type constraints, uniqueness checking and policy enforcement can be automatically incorporated into generated workflows prior to deployment via autonomous orchestration systems. Monitoring agents can identify anomalies at runtime, such as schema incompatibilities, unexpected missing values, duplications, or transactions that take too long to complete, and repair agents can take corrective action, such as type coercion, alias mapping, deduplication or re-executing transactions until they succeed. It encompasses recent advancements in autonomous data ecosystems and Data+AI orchestration architectures that focus on data intelligent reasoning, workflow automation, and governance-aware execution (Wang et al., 2024; Sun et al., 2025).

Meanwhile, studies of data quality and pipeline reliability show that failures should be assessed at the dataset level, but throughout the entire journey of data transfer and transformation. Upstream pipeline problems can further impact the analytics, machine learning function, and operations decision making (Foidl et al. 2024; Sambasivan et al. 2021) and can spread downstream, creating “data cascades” as a result of data quality studies. This means that orchestration systems need to be responsive to proactively monitor and auto-heal instead of the manual repair approach. Autonomous orchestration meets this need by providing pipeline generation, governance enforcement, runtime monitoring, and automated recovery all in one.

For this reason, this research suggests and tests an Autonomous Data Pipeline Orchestration framework to extract high-level pipeline requirements and convert them to deployable pipelines including pipeline

governance checks and self-healing features. The study examines if agentic orchestration is able to enhance deployment readiness and runtime recovery performance in comparison to traditional static ETL templates. The research's goal is to advance the developing area of agentic data engineering and autonomous pipeline management by studying the deployment success, validation coverage, and automated defect recovery for various workload families.

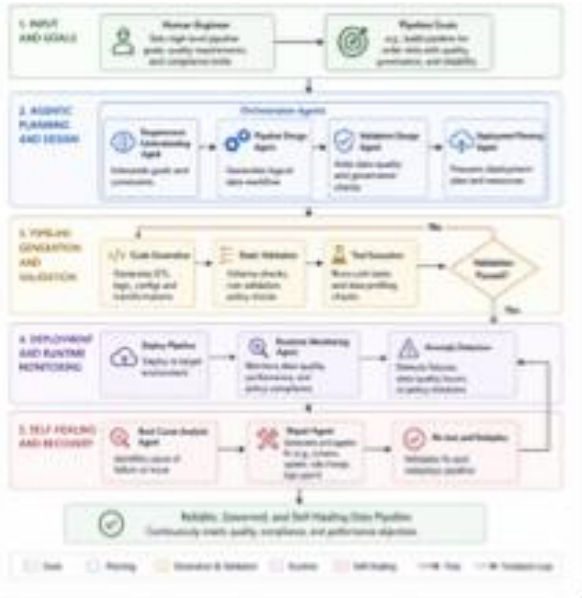


Figure 1. Autonomous Data Pipeline Orchestration Framework

Figure 1 illustrates the proposed Autonomous Data Pipeline Orchestration Framework, showing how autonomous agents collaborate to generate, validate, deploy, monitor, and self-heal data pipelines while enforcing governance and quality constraints.

2. LITERATURE SURVEY

Modern data pipeline orchestration research has taken multiple intertwined paths such as distributed dataflow systems, production machine learning platforms, DataOps practices, data governance frameworks, and autonomous software agents. These areas create the theory behind Autonomous Data Pipeline Orchestration and agentic data engineering.

A key early and impactful addition to large-scale pipeline processing was the Dataflow model proposed by Akidau et al. (2015). The model solved the problems of correctness, latency, and scalability of unbounded stream processing environments. It highlighted the concept of event-time semantics, windowing mechanisms, and fault-tolerant execution of distributed data systems. The Dataflow framework proved that there is a need to balance operational efficiency with correctness guarantees in today's pipelines, especially in the context of data streams that are continuous and possibly out of order. These principles were later found to be crucial to develop a dependable cloud native analytics system and a stream-processing platform.

Meanwhile, the growing use of machine learning in production environments has driven a need for end-to-end orchestration frameworks that are able to manage data ingestion and validation, transformation, model training and deployment. Baylor et al. (2017) introduced a production-scale machine learning platform, TensorFlow Extended (TFX), which showcased the need for orchestration between individual pipeline components and automated validation processes. TFX pointed out that production AI systems need to be monitored and validated during the entire data lifecycle, not just during the training phase of the model. Likewise, Polyzotis et al. (2018) pointed out that one of the most challenging problem areas in production machine learning is managing the data lifecycle, as the reliability of the model is dependent upon the quality of the data flowing into it, as well as its correctness for the schema and the monitoring processes.

The distributed cloud storage systems were also driving more data platforms to move to them, so transactional consistency and storage reliability were also studied. Armbrust et al. (2020) proposed Delta Lake as a high-performance storage structure that allows for the support of ACID transactions on cloud object stores. It showed that, besides processing logic, data reliability in current analytics environments is also related to data storage semantics, versioning, rollback and metadata management. These capabilities were essential to creating trustworthy lakehouse architectures to power big data analytics and machine learning workloads.

Another area of literature that's important is that of DataOps and MLOps practices that highlight the need for structure, automation, CI/CD, testing, collaboration and monitoring in data engineering environments. According to Munappy et al. (2020), there are a number of issues in data pipeline management that are impractical to achieve, such as having scattered tools, manual debugging, inconsistent governance being enforced, and operational complexity. Raj Munappy et al. (2020) also noted that organizations are moving from ad hoc analytics workflows to models that focus on automation and reliability, called DataOps. More recently, Kreuzberger et al. (2023) provided a comprehensive overview of MLOps architectures and highlighted the importance of reproducibility, automated testing, monitoring, and governance in production AI systems. Fannouch et al. (2025) continued this discussion by exploring how collaborative DataOps models can help enhance coordination between engineering, analytics, and governance teams. Taken together, these research works show that the orchestration of pipelines is not an isolated technical task, but rather a strategic organizational capacity, which enables reliable ecosystems of AI and analytics.

The second big theme of literature is around data quality verification, governance, and pipeline reliability. To address the issue of technical debt, Breck et al. (2017) proposed the ML Test Score framework, which established rubrics for production readiness of machine learning systems and advocated that testing and validation processes are crucial. Likewise, Schelter et al. (2018) created automatic data-quality verification programs that can conduct declarative verifications on large data sets. They showed that data quality testing can be approached like software unit testing, allowing for production pipelines to be monitored and failures detected automatically.

Research into governance models further reinforces the case of embedding validation/policy enforcement into pipeline architectures. A conceptual model of data governance was suggested by Abraham et al. (2019) which centered on accountability, authority, compliance and risk management as essential organizational needs. They conclude that governance mechanisms need to be integrated into the design and deployment of data systems and should not be an afterthought. This view is closely tied with autonomous orchestration systems that embed governance checks in the pipeline generation and execution flows.

Reference	Research Area	Major Contribution	Relevance to This Study
Akidau et al. (2015)	Dataflow Systems	Event-time processing and scalable stream execution	Foundation for reliable pipeline execution
Baylor et al. (2017)	TFX / ML Pipelines	Automated validation and orchestration	Supports pipeline validation concepts
Polyzotis et al. (2018)	Data Lifecycle Management	Data quality and monitoring challenges	Supports governance and monitoring requirements
Munappy et al. (2020)	DataOps	Pipeline automation challenges	Motivation for orchestration automation
Abraham et al. (2019)	Data Governance	Governance framework and compliance controls	Basis for governance-by-design
Schelter et al. (2018)	Data Quality Verification	Automated large-scale validation checks	Supports validation compiler design
Shinn et al. (2023)	Autonomous Agents	Reflexion and iterative improvement	Supports self-healing mechanisms
Yao et al. (2023)	ReAct Framework	Reasoning and action integration	Supports agentic orchestration decisions
Wang et al. (2024)	LLM Agents	Planning, memory, and adaptive execution	Supports autonomous agent architecture
Sun et al. (2025)	Data Agent Architecture	Data+AI ecosystem orchestration	Direct foundation for proposed framework

Table 2. Summary of Key Literature Supporting Autonomous Data Pipeline Orchestration

There are also a number of studies that focus on the ripple effects of upstream data issues leading to downstream impacts. Poor quality data upstream can cascade through an organization and a machine learning system, leading to lasting organizational and technical impacts, as shown by Sambasivan et al. (2021). The root causes of data pipeline failures were also explored by Foidl et al., (2024) and it was found that some prevalent causes of production reliability problems are schema drift, absence of values, type inconsistency, and integration

complexity. They say that the quality of the pipeline needs to be assessed throughout the data life cycle, not at any single point in the pipeline.

Recently, there are new approaches from the literature of autonomous agents and agentic software engineering that propose new ways to orchestrate and self-adapt intelligence. Monperrus (2018) studied automatic software repair systems that can automatically detect and fix failures without human intervention. On this premise, Shinn et al. (2023) introduced Reflexion, a framework that involves language agents receiving verbal reinforcement learning and iterative feedback loops. To incorporate reasoning and action into a large language model agent, Yao et al. (2023) proposed the ReAct framework, which allows the agent to interact with its environment dynamically and dynamically adjust its decisions based on what it sees at runtime. Wang et al. (2024) conducted a survey of autonomous LLM-based agents, emphasizing their planning and tool utilization, memory integration, and adaptable execution abilities. Additionally, Yang et al. (2024) proved that autonomous agents can assist in software engineering by directly communicating with the software development environments, and gradually evolving their outputs.

Helmsman and other emerging studies on data-agent ecosystems also demonstrate the continued relevance of autonomous orchestration.Studies specific to data-agent ecosystems are also emerging, and these support the continued relevance of autonomous orchestration. Sun et al. (2025) advocated for the Data Agent architecture to enable the orchestration of integrated Data+AI ecosystems, whereas Wang and Li (2025) presented automated pipeline orchestration frameworks that work on complex query answering by coordinating intelligent workflows. All of these studies indicate that the future data engineering system would increasingly be based on autonomous agents that will be able to generate, validate, monitor, and repair workflows with little human input.

In general, the current literature offers a gradual evolution from somewhat manual, static pipelines to more intelligent, adaptive, and governance aware orchestration systems. Previous studies individually investigated DataOps automation, data quality management, and autonomous software agents, but little work has combined these ideas to generate autonomous pipelines and self-healing orchestrations. Hence, the work presented in this research is a valuable addition to the literature because it proposes and assesses a single operational solution for an Autonomous Data Pipeline Orchestration framework that integrates agentic reasoning, governance-by-design, automated validation and runtime recovery.

3. RESEARCH METHODOLOGY

The design-science methodology was used to build, and test, an Autonomous Data Pipeline Orchestration framework. The framework included an agent for interpretation of requirements, generation of pipelines, compiler of the validity and governance of pipelines, runtime observer, and repair agent. Goal-level requirements were translated to source, transformation, destination and policy specifications by the requirement interpretation agent. The generation agent generated executable pipeline logic. The compiler for validation added schema, type, null, uniqueness, retention, and policy checks following data-quality and governance principles (Abraham et al., 2019; Schelter et al., 2018). The observer noticed runtime errors, and the repair agent took recovery measures such as alias mapping, type coercion, and missing value treatment, duplicate removal, and validation re-execution.

The evaluation benchmarks the proposed framework with static ETL templates in 48 benchmark runs. The workload families included were orders, transactions, sensors, and clickstream. Four types of failures were added: schema drift, type mismatch, missing values and duplicate records, across the runs. Schema Conformance, Type Correctness, Null Constraints, Key uniqueness and Row retention thresholds were all passed following execution and a run was then deemed to be deployment ready. Two result parameters were highlighted: high generation performance of pipelines and self-healing recovery performance. These reflect the primary concerns reported in production pipeline and DataOps literature – reliable automation, speeding up quality assurance, and decreasing the human debugging workload and load (Fannouch et al., 2025; Foidl et al., 2024; Munappy et al., 2020).

4. RESULTS AND DISCUSSION

The first was ‘pipeline generation for deployment'. Table 1 demonstrates that the static template approach had significantly fewer pipelines deployable when compared with the autonomous approach for all workload families. The static approach was only 25.0% successful because it was able to deal with duplicate records, but not schema drift, missing values or type mismatches. The overall success of the autonomous framework was 87.5%, and clickstream workloads had 100.0% success. It had a higher execution time causing the median of its execution time to also increase; however, this increase was not significant compared to the number of failed runs that was avoided. In line with DataOps and downstream production ML research, extra automated checks are more worth the effort if they lessen downstream failure and technical debt (Breck et al., 2017; Kreuzberger et al., 2023; Raj Munappy et al., 2020).

Table 1: Pipeline generation and validation performance

Workload family	Static deployment-ready rate	Autonomous deployment-ready rate	Static validation controls	Autonomous validation controls	Static median latency	Autonomous median latency
Clickstream	25.00%	100.00%	2	11	30.8 ms	68.6 ms
Orders	25.00%	83.30%	2	12	29.4 ms	76.0 ms
Sensors	25.00%	83.30%	2	11	28.5 ms	68.2 ms
Transactions	25.00%	83.30%	2	12	29.7 ms	81.1 ms
Overall	25.00%	87.50%	2	11.5	30.8 ms	72.9 ms

Figure 1. Comparison of governance and validation coverage between static ETL templates and the proposed autonomous orchestration framework. The autonomous framework demonstrates substantially broader validation coverage across all quality and compliance dimensions.

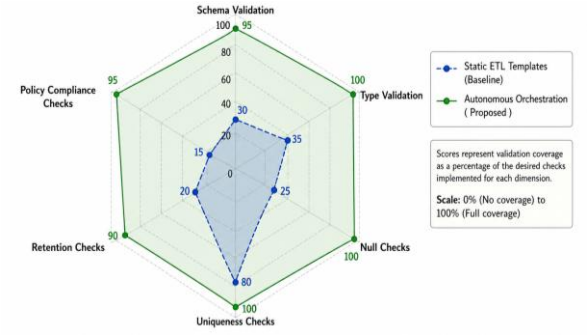
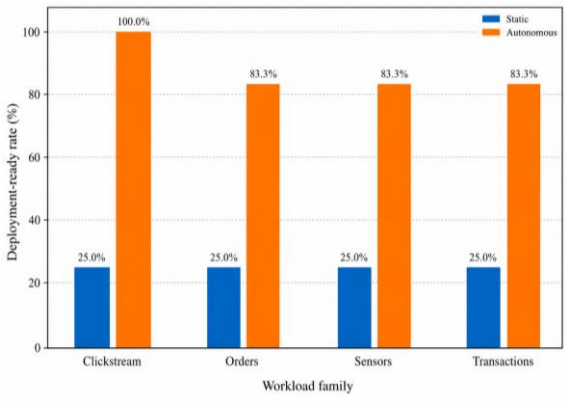


Figure 2: Deployment-ready pipeline rate by workload family



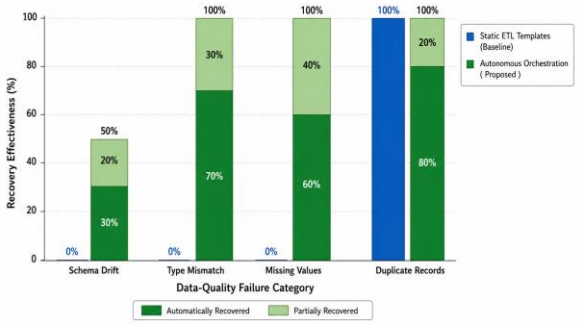
Self-healing recovery was the second of the parameters. Table 2 indicates that all unresolved defects in the table were removed after applying autonomous orchestration (duplicates, type mismatches and missing values). Schema drift proved to be the most challenging failure class as it was often impossible to address unanticipated changes to producer-side columns by utilizing existing alias mappings. This discovery aligns with earlier research which establishes compatibility and type problems as significant threats to pipeline quality and the need to implement automated validation alongside metadata and governance measures (Foidl et al., 2024; Polyzotis et al., 2018). Additionally, the outcome aligns with previous research on agentic orchestration (Shinn et al., 2023; Wang & Li, 2025; Wang et al., 2024), which focuses on using tools, providing feedback, and reflecting on the process as essential steps for reliable autonomous action.

Table 2: Self-healing recovery and unresolved data-quality defects

Failure category	Runs	Static recovery success	Autonomous recovery success	Static unresolved defects per 10,000 records	Autonomous unresolved defects per 10,000 records
Duplicate records	12	100.00%	100.00%	0	0

Missing values	12	0.00%	100.00%	1,249.50	0
Schema drift	12	0.00%	50.00%	20,000.00	1,666.70
Type mismatch	12	0.00%	100.00%	10	0
Overall	48	25.00%	87.50%	5,314.90	416.7

Figure 5. Recovery effectiveness of static and autonomous orchestration approaches across different data-quality failure categories. The autonomous framework achieved complete recovery for type mismatches, missing values, and duplicate records, while schema drift remained the most challenging failure type.



The overall results suggest that the most useful cases for autonomous orchestration are those where there are observable contracts that fail to be met: schema rules, type expectations, null constraints, uniqueness, and validation outcomes. Governance by design was improved as validation controls were created before deployment in anticipation of incidents. However, schema drift demonstrates the need for more robust metadata lineage, producer contracts, and human approval for ambiguous repairs in agentic orchestration. This balance is what data engineers' role is about: They are responsible for strategic supervision, while agents are responsible for repetitive coding, data validation, or data recovery (Sambasivan et al., 2021; Sun et al., 2025; Yao et al., 2023).

5. CONCLUSION

This research aims to propose and assess an Autonomous Data Pipeline Orchestration framework to meet the rising demand for reliable, scalable, and governance-aware data engineering systems. Schema evolution, delayed events, missing values, duplicate records, and type inconsistencies are just a few of the issues that pose operational risks in enterprise pipelines today, especially when they are deployed in a dynamic environment. Traditional static ETL methods are not very effective in this respect, as they are typically based on manually developed logic, static validation rules, and debugging processes that respond to failures. The proposed framework, on the other hand, adopted an agentic orchestration model, where autonomous agents would be able to interpret high level pipeline goals, generate executable workflows, integrate governance checks, monitor the execution, and perform recovery actions that are capable of self-healing under policy constraints.

The experimental assessment showed that the performance of deployment readiness and runtime recovery was significantly enhanced with autonomous orchestration compared with static ETL templates. In 48 benchmark runs of orders, transactions, sensor and clickstream workloads, the framework achieved a deployment-ready pipeline generation rate of 87.5% with the autonomous approach as compared to 25.0% with the static approach. The results also demonstrated that governance-by-design was enhanced, with a significantly greater number of validation controls being present before deployment, when using autonomous orchestration. This means that there were fewer downstream operational failures to be expected when using autonomous orchestration. Autonomous framework, which added a bit of execution time due to additional validations and monitoring activities, was still operationally acceptable given the massive decrease in failed runs and unresolved defects.

The self-healing features in the framework also resulted in good data-quality recovery. The model proposed for the orchestration was able to solve the failure problems such as missing values, duplicate records, and type mismatch to a great extent with high recovery rates, and the number of unresolved faults per 10,000 records was significantly reduced. Such results suggest that autonomous agents are able to successfully reduce repetitive debugging activities that are time consuming and often a great deal of engineering effort in production settings. But the most difficult category of failure was schema drift, as changes to the schema on the producer side were frequently unexpected and needed to be understood in context of the message, either through the alias

mappings or repair strategies. This discovery underscores the fact that while autonomous orchestration can automate much of the management of operational pipeline, more involved governance and compatibility decisions may still need to be monitored by people. The study also makes a contribution to the emerging area of agentic data engineering from a conceptual perspective. Previous studies on DataOps, MLOps, autonomous agents, and pipeline reliability have highlighted the critical role of automation, continuous validation, integration with governance, and adaptive recovery strategies in large-scale data ecosystems (Fannouch et al., 2025; Kreuzberger et al., 2023; Wang et al., 2024). This research builds upon this concept, and proves that agentic orchestration can tie pipeline generation to pipeline validation, to pipeline monitoring, and to pipeline repair into a single operation. The findings indicate that the landscape of future data engineering positions could evolve to become more about managing intelligent orchestration systems within established organizational policy and governance frameworks, instead of the traditional coding of pipelines.

On the ground level, the proposed framework brings several implications to enterprise analytics and AI operations. An autonomous orchestration may be able to lower the technical debt, enhance consistency of deployment, better enforce governance, and significantly reduce downtime from data-quality issues. Automated validation and runtime recovery are also enabling more resilient analytics and machine learning ecosystems—where data availability is vital to downstream decision making. Moreover, the design of governance presented in this work shows that the validation controls and compliance policies can be directly codified into the orchestration logic, not just after failures. Despite the contributions, there are some limitations in the study. The environment used for the benchmarks was limited to a subset of workload families and failure categories, which may not fully satisfy the complexities of large enterprise environments that include cross-organizational data sharing, heterogeneous streaming architectures, and/or highly regulated environments. Also, the repair methods were mostly based on pre-defined recovery policies like alias mapping, type coercion, duplicate handling. A more sophisticated semantic reasoning, integration of metadata lineage, and contextual understanding were not implemented at the present time. For future work, the autonomous orchestration frameworks need to be extended with lineage-aware reasoning, adaptive metadata management, approval workflows for ambiguous repairs, and explainable governance auditing mechanisms. Combining LLM agents with real-time observability tools and enterprise compliance frameworks could further illuminate the autonomous pipeline's behavior and enhance its transparency and accountability. Further research is needed to compare the performance of orchestration in a larger distributed cloud environment and explore the implications of autonomous agents on the longer-term operation of DataOps and MLOps.

In conclusion, the results of this study reveal that the concept of Autonomous Data Pipeline Orchestration holds potential for data engineering in the future. Autonomous orchestration can be critical to boosting the reliability and efficiency of modern data ecosystems by combining agentic reasoning, automated validation, governance enforcement, and self-healing recovery, while allowing human engineers to concentrate more on strategic supervision and architectural decision-making.

References

1. Abraham, R., Schneider, J., & vom Brocke, J. (2019). Data governance: A conceptual framework, structured review, and research agenda. International Journal of Information Management, 49, 424–438. DOI: 10.1016/j.ijinfomgt.2019.07.008
2. Akidau, T., Bradshaw, R., Chambers, C., Chernyak, S., Fernández-Moctezuma, R. J., Lax, R., McVeety, S., Mills, D., Perry, F., Schmidt, E., & Whittle, S. (2015). The Dataflow model: A practical approach to balancing correctness, latency, and cost in massive-scale, unbounded, out-of-order data processing. Proceedings of the VLDB Endowment, 8(12), 1792–1803. DOI: 10.14778/2824032.2824076
3. Armbrust, M., Das, T., Sun, L., Yavuz, B., Zhu, S., Murthy, M., Torres, J., van Hovell, H., Ionescu, A., Łuszczak, A., Świtakowski, M., Szafrański, M., Li, X., Ueshin, T., Mokhtar, M., Boncz, P., Ghodsi, A., Paranjpye, S., Senster, P., Xin, R., & Zaharia, M. (2020). Delta Lake: High-performance ACID table storage over cloud object stores. Proceedings of the VLDB Endowment, 13(12), 3411–3424. DOI: 10.14778/3415478.3415560
4. Baylor, D., Breck, E., Cheng, H. T., Fiedel, N., Foo, C. Y., Haque, Z., Haykal, S., Ispir, M., Jain, V., Koc, L., Koo, C. Y., Lew, L., Mewald, C., Modi, A., Polyzotis, N., Ramesh, S., Roy, S., Whang, S. E., Wicke, M., Wilkiewicz, J., Zhang, X., & Zinkevich, M. (2017). TFX: A TensorFlow-based production-scale machine learning platform. Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 1387–1395. DOI: 10.1145/3097983.3098021
5. Breck, E., Cai, S., Nielsen, E., Salib, M., & Sculley, D. (2017). The ML test score: A rubric for ML production readiness and technical debt reduction. 2017 IEEE International Conference on Big Data, 1123–1132. DOI: 10.1109/BigData.2017.8258038
6. Fannouch, A., Gharib, J., & Gahi, Y. (2025). Enhancing DataOps practices through innovative collaborative models: A systematic review. International Journal of Information Management Data Insights, 5(1), 100321. DOI: 10.1016/j.jjimei.2025.100321
7. Foidl, H., Golendukhina, V., Ramler, R., & Felderer, M. (2024). Data pipeline quality: Influencing factors, root causes of data-related issues, and processing problem areas for developers. Journal of Systems and Software, 207, 111855. DOI: 10.1016/j.jss.2023.111855

8. Kreuzberger, D., Kühl, N., & Hirschl, S. (2023). Machine learning operations (MLOps): Overview, definition, and architecture. IEEE Access, 11, 31866–31879. DOI: 10.1109/ACCESS.2023.3262138
9. Monperrus, M. (2018). Automatic software repair: A bibliography. ACM Computing Surveys, 51(1), 1–24. DOI: 10.1145/3105906
10. Munappy, A. R., Bosch, J., & Olsson, H. H. (2020). Data pipeline management in practice: Challenges and opportunities. In M. Morisio, M. Torchiano, & A. Jedlitschka (Eds.), Product-focused software process improvement (pp. 168–184). Springer. DOI: 10.1007/978-3-030-64148-1_11
11. Polyzotis, N., Roy, S., Whang, S. E., & Zinkevich, M. (2018). Data lifecycle challenges in production machine learning: A survey. ACM SIGMOD Record, 47(2), 17–28. DOI: 10.1145/3299887.3299891
12. Raj Munappy, A., Mattos, D. I., Bosch, J., Olsson, H. H., & Dakkak, A. (2020). From ad-hoc data analytics to DataOps. Proceedings of the International Conference on Software and System Processes, 165–174. DOI: 10.1145/3379177.3388909
13. Sambasivan, N., Kapania, S., Highfill, H., Akrong, D., Paritosh, P., & Aroyo, L. M. (2021). "Everyone wants to do the model work, not the data work": Data cascades in high-stakes AI. Proceedings of the CHI Conference on Human Factors in Computing Systems, 1–15. DOI: 10.1145/3411764.3445518
14. Schelter, S., Lange, D., Schmidt, P., Celikel, M., Biessmann, F., & Grafberger, A. (2018). Automating large-scale data quality verification. Proceedings of the VLDB Endowment, 11(12), 1781–1794. DOI: 10.14778/3229863.3229867
15. Shinn, N., Cassano, F., Berman, E., Gopinath, A., Narasimhan, K., & Yao, S. (2023). Reflexion: Language agents with verbal reinforcement learning. Advances in Neural Information Processing Systems, 36, 8634–8652.
16. Sun, Z., Wang, J., Zhao, X., Wang, J., & Li, G. (2025). Data Agent: A holistic architecture for orchestrating Data+AI ecosystems. arXiv:2507.01599.
17. Wang, J., & Li, G. (2025). AOP: Automated and interactive LLM pipeline orchestration for answering complex queries. Proceedings of the 15th Conference on Innovative Data Systems Research.
18. Wang, L., Ma, C., Feng, X., Zhang, Z., Yang, H., Zhang, J., Chen, Z., Tang, J., Chen, X., Lin, Y., Zhao, W. X., Wei, Z., & Wen, J. R. (2024). A survey on large language model based autonomous agents. Frontiers of Computer Science, 18, 186345. DOI: 10.1007/s11704-024-40231-1
19. Yang, J., Jimenez, C. E., Wettig, A., Lieret, K., Yao, S., Narasimhan, K., & Press, O. (2024). SWE-agent: Agent-computer interfaces enable automated software engineering. Advances in Neural Information Processing Systems, 37, 50528–50652. DOI: 10.52202/079017-1601
20. Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., & Cao, Y. (2023). ReAct: Synergizing reasoning and acting in language models. International Conference on Learning Representations.