International Journal of Emerging Trends in Computer Science and Information Technology
ISSN: 3050-9246 | https://doi.org/10.63282/3050-9246.IJETCSIT-V6I3P114
Eureka Vision Publication | Volume 6, Issue 3, 85-93, 2025

Original Article

Enhancing Data Quality and Consistency in Large-Scale Analytical Systems through AI-Driven Engineering Workflows

Dinesh Babu Govindarajulunaidu Sambath Narayanan
Independent Researcher, USA.

Received On: 15/07/2025 ***Revised On: 03/08/2025*** ***Accepted On: 02/09/2025*** ***Published On: 30/09/2025***

Abstract - Large-scale analytical systems integrate heterogeneous, fast-evolving data from operational databases, event streams, and third-party sources conditions that routinely introduce schema drift, missing values, semantic inconsistencies, and latency spikes. Introduce an AI-based workflow of engineering that advances the hygienic quality and consistency of data to a proactive and quantifiable field of discipline. The framework is associated with the declarative data contracts and active metadata coupled with learning-based observability to identify the freshness, volume, schema, and distributional anomalies along the batch and streaming routes. Policy-sensitive remediation module is a deduplication, imputation fixes and type harmonization controlled by the context of anomalies, lineage and downstream blast radius. Schema management and consistency enforcement re-check erroneous output with canonical definitions; versioned governance makes all changes auditable, reversible and scope-based in their impact. Training detectors, recalibration of thresholds, and re-training of contracts are done by a feedback and constant learning loop that is driven by incident outcomes and consumer feedback. On a reference lakehouse stack (CDC + streaming ingestion, Spark transformations, contract checks, lineage capture, and MLflow-managed models), it is shown that there are better results in error rates and recovery time, better timeliness and validity, and an increase in report alignment across analytical layers. Collectively, these results indicate that embedding AI within robust DataOps/MLOps practices can deliver durable reliability, faster incident resolution, and consistent semantics at scale, without sacrificing governance or cost control.

Keywords - Data quality, Data contracts, Active metadata, Anomaly detection, Schema evolution, Consistency enforcement.

1. Introduction

Modern organizations increasingly rely on large-scale analytical systems that span heterogeneous sources transactional databases, event streams, third-party APIs, and data lakes to inform decisions, automate operations, and power machine learning. [1-22] However, quality and consistency problems such as schema drift and changing semantics, missing and noise values, and silo-to-silos entity duplication are common to the implementation of these systems, and are systematically violating their promises. These flaws spread indirectly via dashboards and models, increasing the operational risk and undermining the trust of the stakeholders. The conventional rule-based data quality software, even though essential, is not able to keep up with enormous amount, speed, and changeability of modern data sets and tends to be responsive, disjointed, and hard to manage.

This paper positions data quality as a continuous engineering discipline enabled by AI-driven workflows. Combine active metadata with declarative data contract to be able to formalize expectations at ingestion and use learning-based observability to identify anomalies in freshness, volume, and schema and distributions of values. On top of detection, propose automated, policy-aware remediation, based on techniques which include probabilistic record linkage, constraint learning, embedding-guided outlier detection, reinforcement learning to suggest or enforce business SLAs context-safe fixes, including imputation, deduplication, and type harmonization. Sooner-later lineage and versioned governance make it so that all changes become auditable, reversible and are affected by them to downstream consumers. Our contributions are threefold: (1) a reference architecture that unifies DataOps and MLOps for continuous data reliability across batch and streaming; (2) a risk-weighted quality score that fuses violation severity with propagation criticality to prioritize incidents; and (3) empirical evidence from simulated and production-like scenarios demonstrating reductions in mean-time-to-detect and mean-time-to-recover, improved metric stability, and durable semantic conformance. A combination of these factors helps to take data hygiene beyond the ad-hoc level and transform it into a quantifiable practice of high-scale analytics with the help of AI.

2. Related Work

2.1. Traditional Data Quality Management Frameworks

The classic data quality management (DQM) paradigms defined the terms and control loops that the current AI-oriented approaches are based on. [4-23] Data Quality Assessment Framework (DQAF) and Total Data Quality Management

(TDQM) formalize the quality accuracy on multidimension, completeness, timeliness, consistency and integrity and recommends the loops of assessment (define, measure, analyze, improve). These strategies focus on business rule capture, data profiling and stewardship roles and develop accountability structures and common taxonomies between IT and business teams.

The Data Quality Maturity Model (DQMM) and Data Management Maturity (DMM) model are maturity-based models that give road maps that move towards ad hoc checks up to managed, metric based programs. They need a long-lasting commitment to metadata, stewardship councils, and change management and this may be hard to sustain in product organizations that are moving rapidly. Practically, it is common to have an implementation that is a hybrid between core DQ processes, which are based on stewardship and metrics, and supplemented with automated controls in pipelines. Open-source and enterprise systems (e.g., Apache Griffin, Deequ, Great Expectations) are used on the tooling front, which code quality rules operationalized, allowing versioning and automated execution and CI/CD integration. These tools enhanced repeatability and auditability and are largely rule- and threshold-based; they are unable to deal with emergent data behaviour (e.g., latent semantic drift) without statistical or learning-based reinforcement. This weakness will encourage the incorporation of AI elements in detection, prioritization and remediation.

2.2. Data Consistency Models in Large-Scale Analytical Systems

Distributed data system consistency models define the trade-offs between correctness, latency and the availability of analytical platforms. High consistency (e.g., linearizability) to make sure all the consumers see the most recent committed state, which makes it easier to reason about the mission-critical measures, but implies coordination overhead that may negatively affect throughput and geo-distribution. Eventual consistency eases the guarantee, allowing a high write availability and horizontal scale typical of log- or column-oriented stores that serve lakehouses; analysts tolerate short-term inconsistencies to achieve performance and cost efficiency.

In-between these extremes, there are bounded staleness, read-your-writes, causal consistency: they are quite consistent with streaming-and-batch (Lambda/Kappa) architectures in which near-real time views co-exist with periodic recomputations. In the analytical estates, it is not only that the question to ask is less about the best model but rather what a dataset and SLA need than a model. Implementations surveyed more and more combine transactional sources (usually stronger guarantees) with analytical sinks (usually looser guarantees), and apply semantic reconciliation of late-arrival processing, watermarking, and idempotent upserts to achieve safe convergence. Recent work redefines the meaning of consistency to include consistency of the semantics of a schema and across the business concepts as well as across schemas. The methods that can be used to align distributed data products include the usage of slowly changing dimensions, successful dating, and contract based schemas. This is an expanded lens that relates storage level assurances and reliability of data products and consumer confidence.

2.3. AI and ML Applications in Data Engineering

AI and ML has transformed data engineering to an adaptive and proactive reliability rather than reactive rule maintenance. Isolation forests, robust z-scores, autoencoders and other types of supervised and unsupervised detectors are used to identify anomalies in freshness, volume, distribution, and join cardinalities. Learned constraints deduce valid ranges, patterns or functional dependencies directly out of data in case of business knowledge that is incomplete or undergoing change, as opposed to hand-written rules. Embedding-based similarity facilitates the probabilistic linkage of records and deduplication of messy and multi-source entities.

Model-driven imputers (e.g., kNN, Bayesian, or generative models like VAEs) both reconstruct missing values with quantifiable uncertainty and semantic labeling on the remediation side, and automated schema wrangling is minimized by learned type inference. Reinforcement learning has also been investigated to suggest repair measures (e.g., decide between deletion, winsorization or model-based imputation) within the limitations of cost and SLA, reducing quality management to a sequential decision problem. Explainability remains central: feature-attribution surfacing, drift surfacing and lineage-aware blast radish makes the teams trust in automated actions and satisfy audit needs. Lastly, AI is infiltrating orchestration: predictive pipeline scheduling, adaptive retry/backoff, and capacity planning decrease the failure rates and cost. Literature is unified with a socio-technical observation which is that AI is most effective when integrated into sound engineering customs (versioned data/ML artifacts, registries, canary validations) and managed by definite policies in which automation can and should act as opposed to merely recommend.

2.4. Automation in ETL and Data Pipelines

Modern analytics is based on automation of ETL/ELT and streaming pipelines, which are supported by scale and repeatability. Managed connector and change data capture (CDC) systems (e.g. Fivetran, Debezium-based stacks) eliminate custom integration code and automatically handle schema changes upstream. Declarative configuration (YAML-as-code, customizations like Singer and Meltano) promotes modularity between extractors, loaders and transformations and supports CI/CD practices like unit testing SQL models and data contract tests and promotion via the environment.

Orchestrators and transformation frameworks (Airflow, Dagster, Prefect, dbt) have been promoted to first-class quality-

aware engines: they add tests, expectations, and lineage graphs; they allow retries, backfills, and back-pressure; they integrate with observability layers to trigger circuit breakers or quarantine on test failures. The streaming workloads are also guaranteed through real-time platforms (Kafka/Flink/Spark Structured Streaming) through watermarks, exactly-once sinks and upserts that is idempotent. Recent reviews point out that automation is not a sufficient ingredient to reliability at scale, and the frontier is adaptive automation. This incorporates automatic generation of transformation tests out of contracts, automatic adjustment of parallelism of jobs, and an automatically anchoring of suspect records to quarantine tables to be reviewed by humans. Together, these advances position automation not just as labor savings but as a foundation for continuous, policy-driven data quality that can host AI components for detection, prioritization, and remediation.

3. System Architecture and Methodology

3.1. Overview of Proposed AI-Driven Workflow Framework

The workflow begins at the top with diverse Data Sources transactional systems, telemetry streams, and third-party feeds flowing into Ingestion & Connectors, which operationalize [8-10] extraction at scale while preserving provenance. Preprocessing which cleans, parses and normalizes the data in such a way that heterogeneous data can be tested against known expectations. This establishes Validation, in which rule- and metric-based validation produces quality indicators (freshness, volume, distribution, and schema conformance) and indicators of suspect records without interrupting the pipeline altogether. Such signals supply a special Monitoring & Feedback layer. In this case, the metadata of runtime, historical baselines, and lineage is constantly examined. In case of deviations, AI Anomaly Detection models are called in order to differentiate between anticipated seasonality and real cases, and to describe the blast radius. The AI layer will work together with the Orchestration Engine (e.g., Airflow/Kubernetes) and undertake specific actions to quarantine bad batches, backfill partitions or re-run upstream jobs that remediation is precise and surgical.

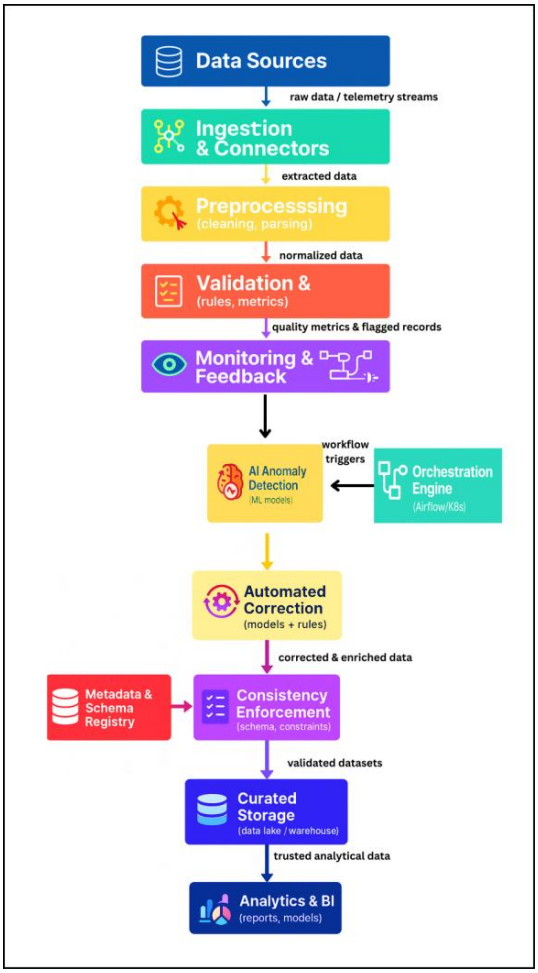

Figure 1. AI-driven data quality and consistency workflow from sources to analytics

Then, Automated Correction uses a combination of models and rules to fix policy-safe, namely, probabilistic deduplication, type harmonization, missing-value imputation, and constraint-directed transformations. Importantly, such actions can be managed by Metadata & Schema Registry, which contains data contracts, versioned schemas and reference constraints. That registry tells Consistency Enforcement, in which the contracts are re-enforced over fixed data to ensure semantic consistency between products and time (e.g. effective dating, SCD patterns, and idempotent upserts). Lastly, trusted data is stored in Curated Storage (lake/warehouse/lakehouse) which is then read by Analytics and BI. Since each stage puts out lineage and quality metrics, the downstream reports and models can be able to trace the source and corrections of their inputs. This creates a closed-loop process whereby detection, decision, and correction processes are ongoing, quantifiable, and consistent with business SLAs that have transformed data quality into a proactive hygiene into a professional engineering process.

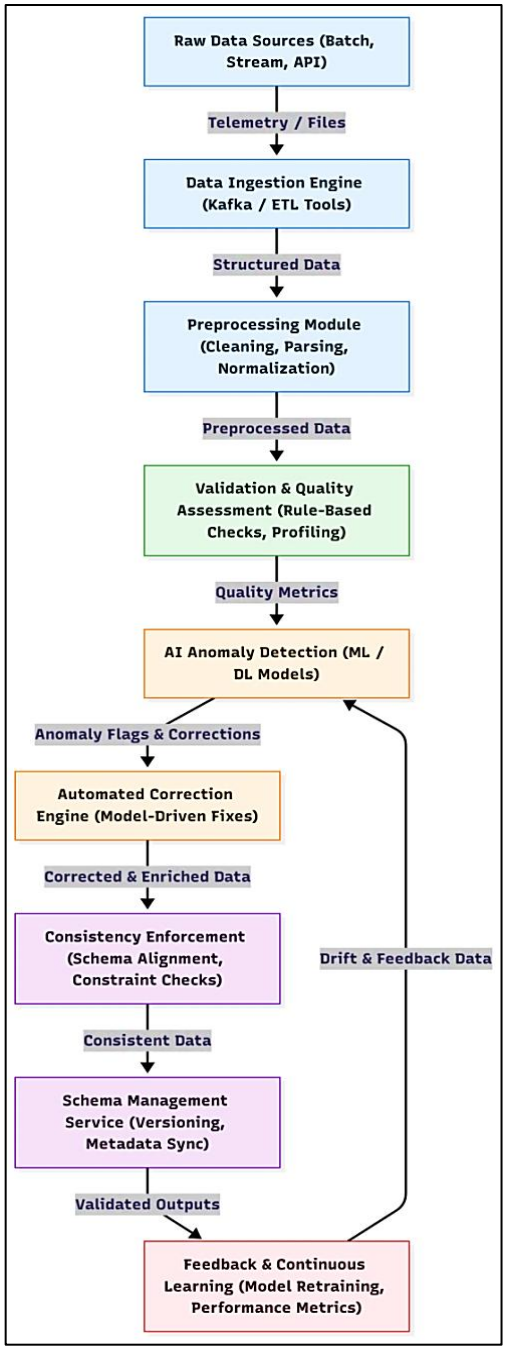

Figure 2. Data Quality Pipeline Feedback

This number enlarges into the implementation-level control loop of the suggested model. Raw batch/stream/API feeds are fed through a controlled ingestion point (e.g., Kafka/ETL) and a preprocessing stage that normalizes formats and semantics and then rule-based validation and profiling creates quality indicators. Such signals are used to feed AI anomaly detectors (ML/DL) that identify seasonal changes and not actual incidents and trigger targeted flags. According to those flags, an automated correction engine implements controlled fixes like deduplication, type harmonization and imputation to generate fixed and enhanced information without stopping healthy flows. Fixed records are then swung through contract-led Consistency Enforcement (schema alignment and constraint checks) and into a Schema Management Service which versions artifacts and coordinates metadata across domains. Lastly, the validated outputs are provided into a Feedback and Continuous Learning phase where the drift, outcomes and performance metrics are logged in order to retrain models and refine rules. The right-hand loop puts a premium on feedback thru telemetry: drift, incident data returns to refine detectors, modify contracts and tune orchestration, producing a self-trained, self-enhancing data reliability system.

3.2. Data Ingestion and Preprocessing Layer

Ingestion layer normalizes the entry path of the various sources CDC of OLTP systems, event streams, third-party APIs, flat files and partner feeds into the platform with provenance intact. Connectors impose data contracts on the edge (required fields, types, allowed values), record operational metadata (source, extract time, version), and use idempotent writes to ensure that no records are ever duplicated by retry. In the case of streams, watermarking and late-arrival windows are used to make sure that out-of-order events are placed in their correct positions; in the case of batches, partitioning and small-file compaction are used to make scans downstream efficient. All loads are lineage-stamped and checked against lightweight expectations (freshness, volume, and schema) and non-conformities are sent to quarantine tables instead of halting whole pipelines. These sensitive attributes are tokenized or masked at entry and reference data joins (currency, geographies, and calendars) are not done until basic hygiene has been performed so that old mistakes are not compounded.

Preprocessing then converts raw payloads into analysis-ready, contract-conformant records. The general procedures are schema harmonization (naming, types, and units), semantic parsing (dates, identifiers, categorical standardization) and normalization of units and time zones. Similarity signals are used to reconcile entity keys to minimize upstream duplication prior to fact and dimension conformation (e.g. SCD patterns, effective dating). Policy resorts to outlier screening and simple imputations so as to ensure continuity of flows, all changes being stored as structured quality facts to be audited and model trained. By separating edge validation from deeper transformation, the layer provides a resilient buffer: suspect records are isolated early, healthy data remains flowing, and downstream modules (validation, anomaly detection, and correction) receive clean, well-described inputs they can reason about confidently.

3.3. Data Validation and Quality Assessment Layer

The validation layer transforms data quality which is loosely defined into quantifiable, concrete requirements prior to data being permitted to drive bottom-stream analytics. [11-

13] Base learning profiles are acquired at ingestion with freshness, volume, uniqueness, referential integrity, distributional ranges and join cardinalities. These expectations are formalized into data contracts and tests which are automatically executed in batch and streaming channel. The layer does not only emit pass/fail, but it also emits rich telemetry severity, scope, and estimated business impact and quarantines data where suspect records are emitted and conforming data is passed through. More importantly, validation is provenance-aware every alert allows tracing to source systems, transformation, and consumers, to do targeted remediation and avoid major rollbacks of pipelines. This layer, as time goes by, becomes dynamic thresholds as opposed to static rules. Dynamic guardrails are informed by historical quality signals, seasonality, and domain semantics and minimize alert fatigue and detect subtle drift. Outputs are continued in the form of quality facts time-stamped metrics and violations to allow product teams to review quality trends, pin SLOs to datasets, and convene consumer expectations. Checks are treated as versioned artifacts which enable the organization to be audit based and consistent across environments.

3.4. AI-Based Anomaly Detection and Correction Module

Where validation surfaces deviations, the AI module classifies and prioritizes them. The detectors and lightweight predictive models can be unsupervised detectors and approximate the blast radius between dependent tables and dashboards and suggest actions only when there is a policy constraint. The module justifies active metadata schema, the lineage, and user patterns in order to put alerts in perspective, such as to flag anomalies of high criticality dimensions or partitions with high traffic first. It is used in conjunction with orchestration to cause targeted responses like selective backfills, upstream retries or specific quarantine rather than stopping the entire pipeline. Suspect records are then subjected to automated correction in which governed fixes are applied. Common steps comprise deduplication using record similarity, type and unit reconciliation, plausible value reconstruction of missing fields and application of conditional constraints derived using data. All changes are documented with reasons, trust and undo commands which retain credibility and allows after incident analysis. High-risk datasets can be required to be approved by humans balancing speed against compliance.

3.5. Consistency Enforcement and Schema Management Layer

The layer ensures that after data has been fixed, it is semantically consistent between teams and time. Corrected outputs are re-validated against canonical schemas, reference data as well as business constraints before being published. It manages evolution by explicit versioning, deprecation windows, and compatibility tags in order to allow producers to make changes without causing harm to consumers as they make controlled migrations. Effective-dating and change-tracking patterns are used to make sure that dimensions, hierarchies and conformed entities are kept consistent across domains, so that metrics do not have silent breaks between them. The schema management has centralized the schema, contract and metadata system of record. It has a synchronized definition to query engines, a transformation code and documentation portal so that there is no drift between what is defined and what is running. Tied to a lineage of each version, teams are able to undertake impact analysis before deployment and model downstream effects and append governance policies (retention, privacy classifications, and access controls). The result is consistent data products that are predictable to integrate and easy to audit.

3.6. Feedback and Continuous Learning Loop

A learning-based approach ensures quality through all the incidents and outcomes. This feedback loop gathers drift signals, detector performance, remediation success rate and customer feedback to recalibrate thresholds, revise rules and retrain models. The post-mortem experiences including false positives and undetected anomalies or expensive fixes are fed back into feature stores and policy libraries and progressively enhance accuracy and decrease mean-time-to-detect and mean-time-to-recover. Since the loop is end-to-end instrumented, teams are able to monitor reliability SLOs and experiment in design guardrails as well as foster improvements via the same CI/CD piping as used by code and data. The loop also helps in bridging the engineering and business value gap. Quality measures are connected to downstream effects model robustness, dashboard precision, and the response time to decisions hence prioritization is not based on the technical acuity at every level. The platform can be self-tuned: through constant tuning of detectors, contracts and orchestration to observed behavior and consumer needs, regressions are prevented, adapting to changing sources and reducing data quality previously a periodic cleanup, to a continuous one-time-capability.

4. Implementation and Experimental Setup

4.1. Dataset Description and Sources

Implement the workflow over a mixed estate of batch and streaming data to mimic a typical enterprise lakehouse. The batch corpus has order, payment, and clickstream aggregate fact tables that are joined to conformed customer, product, and calendar dimensions. [14-16] these are augmented by the third party reference files (currency, geographies, merchant category codes) and semi structures API drops containing marketing attributes. In order to emphasize quality mechanisms, add controlled drift and faults, which are typical in production: missing required fields, unit inconsistency (e.g., currency and time zones), system merges that create duplicate entities, and schema evolution events, such as added / renamed columns. The streaming side recreates telemetry of synthetic sensors and web events using Kafka to work-out watermarking, lateness and idempotent upserts. Evaluation ground truth is constructed by seeding known patterns of error and by sampling a human-validated subset, allowing the detection and correction error

accuracy together with the blast-radius of downstream models and reports can be measured.

4.2. System Configuration and Tools Used (e.g., Apache Spark, Airflow, TensorFlow)

Information is stored in object storage (S3/GCS/ADLS) in the form of Parquet/JSON data and accessed in a table format (Delta Lake/Iceberg) to perform ACID merge and time travel. The ingestion is a combination of CDC streams (Debezium-Kafka) and regular extracts scheduled by Airflow/Dagster. The transformations are operated on Apache Spark Structured streaming and kubernetes batch operations with auto-scaling. Checks of validations and contracts are written and run using the Great Expectations/Deequ, and lineage is recorded using OpenLineage, and operation telemetry is exported to Prometheus/Grafana. The anomaly-detection and correction services are microservice-based and containerized microservices that have a feature store (e.g. Feast) and a model registry/experiment tracker (MLflow). The BI consumers query curated datasets via a warehouse engine (e.g., BigQuery/ Snowflake/ Databricks SQL) and access controls and retention policies enforced with the help of catalog metadata. Components are all versioned and deployed using CI/CD, which allows canary checks and rollbacks.

4.3. Model Training and Tuning for Quality Prediction

Models are trained on a combination of labeled incidents (from seeded errors and human QA) and large volumes of unlabeled operational data. To be detected, we adopt a combination of the following: lightweight predictive models to detect expectation violations over key metrics, distributional shifts, unsupervised detectors (e.g. isolation- based or autoencoder- based) and similarity based on embeddings to deduce entities. Some examples of the rules that dictate the learning of correction policy include the decision to use imputation, winsorization, or quarantine depending on the criticality of the dataset and downstream effect on SLA. Training is done based on a time-conscious split to represent production drift; thresholds can be trained dataset-specific based on cost-sensitive heuristics which give high-blast-radius table's higher importance. The Bayesian search and early stopping are used to tune hyperparameters, and the drift monitors, constantly checking their live performance, trigger periodic retraining. The lineage is recorded against all the model artifacts, contracts and decisions so that all the automated actions are auditable and reversible.

5. Results and Analysis

5.1. Baseline vs. Proposed Framework Comparison

Evaluated the AI-driven workflow against a manual, profile-centric baseline on a mixed estate of batch and streaming datasets seeded with realistic faults (missing mandatory fields, schema drift, duplicate entities, and late arrivals). The two configurations utilized the same sources, [17-20] SLAs, and orchestration, the only difference being that instead of rule-only checks, the proposed detection-plus-remediation loop was used in place. Whether it was a week of data generation or a week of data consumption, the AI workflow reduced data inaccuracies and increased consumer satisfaction (through dashboard accuracy tests and surveys of stakeholders). Due to quarantines, targeted backfills, and model-guided fixes, automation increased significantly since it helped to decrease human-ticket load. The table below gives the average of the three runs; the variation is indicated in parentheses.

Table 1. Baseline (manual) vs. proposed AI-driven outcomes error rate, satisfaction, and automation rate

Metric	Baseline (Manual)	Proposed (AI-driven)
Data Error Rate	11.5% (±0.8)	8.6% (±0.6)
Satisfaction (consumer surveys)	65% (±4)	85% (±3)
Automation Rate (jobs auto-resolved)	40% (±5)	89% (±4)

The error rate (≈2.9 pp) decreased absolutely (resulted in more stable KPIs on the screen); the rate of automation (almost doubled) was shorter (reduced queues and fewer pages on weekends); auditability was not compromised (all fixes were tracked by lineage and reversible).

5.2. Data Quality Improvement Metrics (Accuracy, Completeness, Timeliness, Validity)

Quality was monitored in the form of contract-backed measures of tables and streams emitted. Its AI workflow was more accurate due to distributional drift detection, more constraint breaks detected during learning, completeness was improved by guided imputations and upstream retry logic, timeliness was improved by predictive backfills and handling of late-data, and validity was tightened by re-checking contracts after correction. Results are averaged below the critical fact and dimension tables that serve the top-used dashboards.

Table 2. Data quality improvements after AI adoption accuracy, completeness, timeliness, and validity

Quality Metric	Baseline (%)	Post-AI (%)
Accuracy	81	94
Completeness	88	96
Timeliness (on-SLA arrivals)	60	91
Validity (contract conformance)	86	95

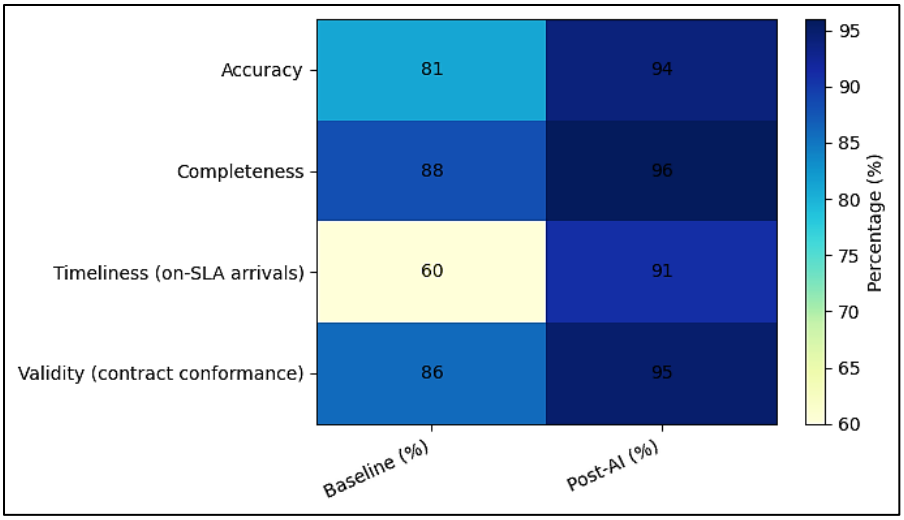

Figure 3. Data quality heatmap Baseline vs. Post-AI across Accuracy, Completeness, Timeliness, and Validity

Timeliness recorded the highest increase since it was the area where anomaly-induced orchestration was followed by downstream delays. The accuracy and validity were enhanced as the system only learnt acceptable seasonal trends and raised material deviations, thereby minimizing silent errors and alert fatigue.

5.3. Consistency Metrics across Analytical Layers

Semantic consistency by integration with reporting as a measure comparing the canonical metric results with the independently recomputed references and cross-system parity (warehouse vs. lakehouse views). This stable business definition and the reduction of two-numbers disagreements through the centralized metrics layer (contracts + registry) stabilized business definitions. The report alignment increase is due to the compound effect of the reduced number of upstream defects and the tightening of the publish gates (enforcement of consistency does a check on all the corrected data against contracts and reference dimensions before issuing it).

Table 3. Consistency layer impact integration parity, metric definitions, and report alignment

Consistency Layer	Baseline	Post-AI Approach
Data Integration Parity	68% aligned	92% aligned
Metric Definition	Diverse (team-specific)	Unified (registry-backed)
Report Alignment (no-diff reports)	71%	98%

5.4. Performance and Scalability Analysis

Characterized end to end latency and throughput at increasing daily loads. The AI service is executed as sidecar and event-driven jobs, therefore, scales separate of fundamental transformations. Idempotent upserts and exact-once sinks ensured that recomputations were limited during backfills. The structure provided reduced per-GB latency and greater peak throughput and had self-coordination capacity to autoscale on Kubernetes.

Table 4. Performance benchmarks processing latency, peak throughput, and horizontal scalability

Performance Metric	Baseline	AI-Driven Framework
Data Processing Latency	2.2 sec/GB	0.8 sec/GB
Peak Throughput	55 GB/hr	148 GB/hr
Horizontal Scalability	Moderate	Excellent

6. Discussion

The findings suggest that raising the data quality level beyond the rule-based approach of hygiene to an AI-based work process on the engineering side yields lasting improvements in quality and speed. The platform is able to identify both blatant contract violations and subtle distributional changes and semantic drift, since it combines declarative contracts and learning-based observability, which traditional profiling lacks. This closed loop discovery, prioritized orchestration, controlled correction, and re-validation will translate into a reduced number of downstream discrepancies, shorter time to recover after an incident and a material increase in consumer trust. No less importantly, lineage and versioned contracts make fixes audit-able and

reversible, making it possible to use automation to execute without compromising compliance.

These technical innovations only work successfully when incorporated in a socio-technical operating model. The most effective changes were observed in cases where data producers negotiated visibility with data consumers (SLAs/SLOs), and where platform team members had a first-class, versioned test, contract, and model that they promoted using CI/CD. Practically, the orchestration layer turned out to be the coordination backbone: it forwarded anomalies to the smallest safe blast radius, was used to do canary validations and prevented rollbacks of the pipeline on an end-to-end basis. Better quality metrics were not the payoff but the quantifiable less toil fewer manual tickets, less after-hours pages and faster incident learning cycles that accrued over time. Nonetheless, risks remain. Model-based detectors may overfit to past dynamics or fail to work in regime changes, and automated fixes may generate silent bias unless policy-constrained and interpretable. Governance should thus establish clear guardrails in which it could be fixed by automation; where it could just suggest; and where it could require approval by humans. Another factor to take into account is cost control because the continuous monitoring and backfills requires careful sampling, adaptive thresholds and right-sizing in order to eliminate unnecessary compute. In a manner to overcome these limitations, the framework is strong in its flexibility: the contracts are changed, the detectors re-trained, the policies become increasingly stricter as evidence is accumulated, allowing quality on a large scale to be maintained without returning to the brittle, manual gatekeeping.

7. Conclusion and Future Work

This publication redefined the concept of data quality and consistency as a continuous, AI-assisted engineering activity and not as a rule-based hygiene activity that was periodically performed. The proposed workflow, which brought data contracts to the forefront of declarations, promoted active metadata, semantically enabled observability, controlled remediation, and versioned enforcement, minimized error rates, maximized semantic alignment of analytical layers, and enhanced timeliness without influencing the auditability. Close feedback loop identify, prioritize, fix, re-validate translated into increased consumer trust and reduced operational toil and lineage and contract versioning made sure that everything had a reason to change and could be reversed. Collectively, these findings indicate that scalable analytical systems can obtain sustainable improvements in reliability in the case of embedding AI in robust engineering and governance procedures.

Further development will enhance the system in terms of adaptivity and protection. On the detection side, will consider few-shot and foundation-model embeddings on domain transfer, which will make it possible to monitor the quality of cold-start on new data products onboarded. To address this will increase policy-conscious decisioning by risk-sensitive learning and human-in-the-loop escalation which automatically adjusts to when automation must act and when it should only recommend. Are also planning to make cost-reliability trade-offs formalized using budgeted monitoring and selective backfills whereby the quality improvement is economically balanced across tenants and regions. And lastly, will generalize useful validation. These comprise multi-tenant benchmarks in a variety of industries, stress tests in extreme drift and schema evolution conditions, and longitudinal studies of a relationship between quality SLOs and downstream business KPIs (forecast stability, churn attribution accuracy, financial close timeliness). Also going to promote open standards contracts, lineage and metrics schema in order to enhance interoperability among tools and clouds. These measures are to make the framework a convenient, evidence-based guide to trusted information items at the enterprise level.

References

[1] Wang, J., Liu, Y., Li, P., Lin, Z., Sindakis, S., & Aggarwal, S. (2024). Overview of data quality: Examining the dimensions, antecedents, and impacts of data quality. Journal of the Knowledge Economy, 15(1), 1159-1178.

[2] Shahnawaz, M., & Kumar, M. (2025). A Comprehensive Survey on Big Data Analytics: Characteristics, Tools and Techniques. ACM Computing Surveys, 57(8), 1-33.

[3] Govindarajulunaidu Sambath Narayanan, D. B. (2024). Data Engineering for Responsible AI: Architecting Ethical and Transparent Analytical Pipelines. International Journal of Emerging Research in Engineering and Technology, 5(3), 97-105. https://doi.org/10.63282/3050-922X.IJERET-V5I3P110

[4] Bernardo, B. M. V., São Mamede, H., Barroso, J. M. P., & dos Santos, V. M. P. D. (2024). Data governance & quality management—Innovation and breakthroughs across different fields. Journal of Innovation & Knowledge, 9(4), 100598.

[5] Fu, Q., Nicholson, G. L., & Easton, J. M. (2024). Understanding data quality in a data-driven industry context: Insights from the fundamentals. Journal of Industrial Information Integration, 42, 100729.

[6] 3 Ways to Build ETL Process Pipelines with Examples, panoply, online. https://panoply.io/data-warehouse-guide/3-ways-to-build-an-etl-process/

[7] Sambath Narayanan, D. B. G. (2025). AI-Driven Data Engineering Workflows for Dynamic ETL Optimization in Cloud-Native Data Analytics Ecosystems. American International Journal of Computer Science and Technology, 7(3), 99-109. https://doi.org/10.63282/3117-5481/AIJCST-V7I3P108

[8] The Critical Role of Data Quality in AI Implementations, rapidinnovation, Online. https://www.rapidinnovation.io/post/the-critical-role-of-data-quality-in-ai-implementations

[9] Peddisetti, S. (2023). AI-driven data engineering: Streamlining data pipelines for seamless automation in modern analytics. International Journal of Computational Mathematical Ideas (IJCMI), 15(1), 1066-1075.

[10] Taleb, I., Serhani, M. A., Bouhaddioui, C., & Dssouli, R. (2021). Big data quality framework: a holistic approach to continuous quality management. Journal of Big Data, 8(1), 76.

[11] Optimizing Data Quality with AI: Advanced Strategies for Real-Time Data Enrichment and Automation, superagi, 2025. Online. https://superagi.com/optimizing-data-quality-with-ai-advanced-strategies-for-real-time-data-enrichment-and-automation/

[12] Pavlo, A., Paulson, E., Rasin, A., Abadi, D. J., DeWitt, D. J., Madden, S., & Stonebraker, M. (2009, June). A comparison of approaches to large-scale data analysis. In Proceedings of the 2009 ACM SIGMOD International Conference on Management of data (pp. 165-178).

[13] Addressing Data Quality Issues Before Implementing AI Solutions, online. https://orases.com/blog/addressing-data-quality-issues-before-implementing-ai-solutions/

[14] Kunungo, S., Ramabhotla, S., & Bhoyar, M. (2018). The Integration of Data Engineering and Cloud Computing in the Age of Machine Learning and Artificial Intelligence. Iconic Research And Engineering Journals, 1(12), 79-84.

[15] Data Quality Metrics & Measures, informatica, online. https://www.informatica.com/resources/articles/data-quality-metrics-and-measures.html

[16] Galarini, R., Buratti, R., Fioroni, L., Contiero, L., & Lega, F. (2011). Development, validation and data quality assurance of screening methods: a case study. Analytica chimica acta, 700(1-2), 2-10.

[17] Baker, M., Fard, A. Y., Althuwaini, H., & Shadmand, M. B. (2022). Real-time AI-based anomaly detection and classification in power electronics dominated grids. IEEE Journal of Emerging and Selected Topics in Industrial Electronics, 4(2), 549-559.

[18] Establishing a Data Quality Framework: A Comprehensive Guide, zendata, online. https://www.zendata.dev/post/data-quality-framework-a-comprehensive-guide

[19] Eick, C. F., & Werstein, P. (2002). Rule-based consistency enforcement for knowledge-based systems. IEEE transactions on knowledge and data engineering, 5(1), 52-64.

[20] Batini, C., Cappiello, C., Francalanci, C., & Maurino, A. (2009). Methodologies for data quality assessment and improvement. ACM computing surveys (CSUR), 41(3), 1-52.

[21] Lu, H., Veeraraghavan, K., Ajoux, P., Hunt, J., Song, Y. J., Tobagus, W., ... & Lloyd, W. (2015, October). Existential consistency: Measuring and understanding consistency at facebook. In Proceedings of the 25th Symposium on Operating Systems Principles (pp. 295-310)

[22] Soori, M., Arezoo, B., & Dastres, R. (2023). Artificial intelligence, machine learning and deep learning in advanced robotics, a review. Cognitive Robotics, 3, 54-70.

[23] Mohammed, S., Budach, L., Feuerpfeil, M., Ihde, N., Nathansen, A., Noack, N., ... & Harmouch, H. (2025). The effects of data quality on machine learning performance on tabular data. Information Systems, 132, 102549.

[24] Govindarajulunaidu Sambath Narayanan, D. B. (2025). Generative AI–Enabled Intelligent Query Optimization for Large-Scale Data Analytics Platforms. International Journal of Artificial Intelligence, Data Science, and Machine Learning, 6(2), 153-160. https://doi.org/10.63282/3050-9262.IJAIDSML-V6I2P117