

Original Article

Design and Evaluation of an Automated Data Quality and Governance Framework for Large-Scale Enterprise Data

Harshita Reddy, Kotla

Independent Researcher, USA.

Received: 07-12-2026

Revised: 07-31-2026

Accepted: 08-13-2026

Published: 08-16-2026

ABSTRACT

The growing volume, diversity, and velocity of enterprise data create challenges in maintaining data quality, consistency, traceability, and governance. Conventional approaches often depend on fragmented validation rules, manual inspection, and isolated governance mechanisms, limiting continuous quality management at scale. This study proposes an automated data quality and governance framework integrating continuous data profiling, quality validation, anomaly detection, automated remediation, governance-policy enforcement, metadata management, and data lineage within a unified data engineering architecture. The framework identifies missing values, duplicates, invalid records, schema violations, inconsistencies, and anomalous observations across enterprise data sources. An experimental evaluation uses controlled datasets containing systematically introduced quality defects. Performance is assessed using precision, recall, F1-score, data-quality improvement, remediation success rate, governance compliance, lineage completeness, and throughput, latency, and resource utilization. Simulated results indicate effective defect detection and remediation, strong governance and lineage coverage, and scalable processing performance. The framework provides a reproducible foundation for trustworthy, scalable, and maintainable enterprise data platforms.

KEYWORDS

Data Engineering, Data Quality, Data Governance, Automated Data Validation, Data Lineage, Enterprise Data Management.

1. INTRODUCTION

1.1. Background and Motivation

The rapid expansion of digital transformation has resulted in the generation of increasingly large, diverse, and continuously changing volumes of enterprise data. Modern organizations collect information from transactional databases, enterprise applications, application programming interfaces (APIs), cloud platforms, Internet of Things devices, analytical systems, and external data sources. These environments have created complex data ecosystems in which information must be integrated, processed, governed, and delivered to downstream analytical and operational applications. As data environments become more distributed, maintaining reliable and trustworthy information has become a fundamental requirement for effective data engineering and organizational decision-making.

Data quality is multidimensional and encompasses characteristics such as completeness, validity, consistency, uniqueness, accuracy, and timeliness (Pipino et al., 2002; Wang et al., 2024). Poor-quality data can propagate through data pipelines and subsequently affect analytics, reporting, machine learning, and operational processes. Recent research indicates that data-quality problems can originate at different stages of data pipelines, including data ingestion, integration, transformation, and processing (Foidl et al., 2024). Consequently, data quality should not be treated solely as a downstream inspection activity but as a continuous engineering concern throughout the data lifecycle.

The increasing importance of trustworthy data has also strengthened the relationship between data quality and data governance. Data governance establishes organizational mechanisms for managing data ownership, policies, metadata, accountability, and appropriate data use, while data-quality processes provide mechanisms for assessing whether data satisfy predefined requirements. Recent literature emphasizes the growing integration of data governance and quality management as organizations seek more reliable and accountable data ecosystems (Bližnák et al., 2024; Bernardo et al., 2024).

1.2. Enterprise Data Quality Challenges

Large-scale enterprise data environments are exposed to numerous quality problems. Missing and incomplete records can reduce the usability of datasets and introduce uncertainty into analytical processes. Completeness is therefore a fundamental dimension of data quality, particularly when different sources contain varying levels of information (Daraio et al., 2022; Timmerman et al., 2023).

Duplicate records represent another important challenge, particularly when information from multiple systems is integrated. Entity-resolution techniques are commonly required to identify records referring to the same underlying entity and prevent duplication from propagating through analytical datasets (Simonini et al., 2022). Invalid values, incorrect data types, and inconsistent representations can similarly reduce data validity and consistency.

Schema inconsistencies create additional challenges when data are integrated across heterogeneous systems. Differences in field names, data types, structures, and representation

conventions can complicate automated processing and increase the need for data transformation and validation. Furthermore, anomalous records may remain undetected when organizations rely exclusively on predefined validation rules. Research on data-stream quality control highlights the importance of continuous mechanisms capable of identifying quality problems as data are generated and processed (Mirzaie et al., 2023).

These challenges are particularly important in large-scale pipelines because errors can propagate rapidly across downstream systems. Data-quality problems therefore require mechanisms that combine automated detection, monitoring, remediation, and traceability rather than relying on isolated validation activities.

1.3. Limitations of Existing Approaches

Existing approaches provide important foundations for data-quality management, but several limitations remain. Traditional data-quality assessment frequently depends on predefined rules, manual inspection, or periodic quality assessments, which can become difficult to maintain as data volume and system complexity increase (Chandran & Gupta, 2022). Automated approaches have improved the ability to profile, validate, and clean datasets, but these capabilities are often implemented as separate components rather than as an integrated data engineering workflow.

Recent research has explored automated exploratory analysis, data cleaning, active learning, and machine-learning-assisted quality control (Patel et al., 2023; Li et al., 2024; Mecca et al., 2024). However, automated remediation remains challenging because corrective actions must preserve data semantics and avoid introducing additional errors. At the same time, data governance research has emphasized catalogs, metadata, lineage, and organizational policies, but these capabilities are not always tightly integrated with continuous quality assessment (Jahnke & Otto, 2023; Chen et al., 2024).

Another limitation concerns scalability. Many quality approaches demonstrate effectiveness on specific datasets or individual quality tasks without comprehensively evaluating throughput, latency, resource utilization, and performance under increasing workloads. This creates a need for an integrated framework that evaluates data quality and governance as part of a scalable data engineering architecture.

1.4. Research Gap

The literature therefore reveals a gap between automated data-quality management and enterprise-scale data governance. Existing studies commonly address individual components such as quality assessment, data cleaning, anomaly detection, governance, metadata, or lineage. However, comparatively less attention has been given to a unified architecture that combines continuous quality assessment, automated remediation, governance-policy enforcement, lineage, observability, and scalable processing within one experimentally evaluated framework.

1.5. Aim and Objectives

The aim of this study is to design and evaluate an automated data quality and governance framework for large-scale enterprise data.

The specific objectives are to:

- Design an integrated architecture for automated data-quality assessment and governance.
- Develop automated mechanisms for detecting common enterprise data-quality problems.
- Implement automated remediation and revalidation processes.
- Integrate governance, metadata, and lineage capabilities.
- Evaluate detection accuracy, remediation effectiveness, and governance compliance.
- Assess scalability, latency, and resource utilization under increasing workloads.

1.6. Research Questions

The study addresses six research questions:

- RQ1: How effectively can the proposed framework detect data-quality problems in large-scale enterprise datasets?
- RQ2: How accurately can automated anomaly-detection mechanisms identify previously undefined quality problems?
- RQ3: To what extent can automated remediation improve overall data quality?
- RQ4: How effectively can integrated governance and lineage mechanisms improve data traceability and policy compliance?
- RQ5: How does the proposed framework perform as data volume and processing workload increase?
- RQ6: What computational overhead is introduced by continuous data-quality and governance monitoring?

1.7. Research Contributions

This study makes five principal contributions. First, it proposes an integrated data engineering architecture combining automated quality assessment, anomaly detection, governance, remediation, lineage, and observability. Second, it provides a quality-aware evaluation framework covering multiple data-quality dimensions and detection metrics. Third, it integrates automated remediation with post-remediation validation to reduce persistent quality defects. Fourth, it evaluates scalability using progressively larger workloads and distributed processing configurations. Finally, it provides a reproducible experimental framework for assessing the interaction between data quality, governance, and large-scale data engineering performance.

1.8. Paper Organization

The remainder of this paper is organized as follows. Section 2 reviews existing research on data quality, automated cleaning, data pipelines, governance, metadata, and lineage. Section 3 presents the proposed automated data quality and governance framework. Section 4 describes the experimental datasets, implementation environment, evaluation metrics, and benchmark procedures. Section 5

presents the experimental results. Section 6 discusses the findings and their implications. Section 7 identifies future research directions, while Section 8 concludes the study.

2. BACKGROUND AND RELATED WORK

2.1. Data Quality Concepts and Dimensions

Data quality refers to the degree to which data are suitable for their intended operational, analytical, or decision-making purposes. It is inherently multidimensional because data can be reliable in one dimension while deficient in another. Foundational data-quality research identifies dimensions including completeness, accuracy, consistency, and timeliness as important characteristics for evaluating information quality (Pipino et al., 2002; Batini & Scannapieco, 2006).

Completeness concerns the presence of required data elements. Missing values can reduce the analytical usefulness of datasets and may introduce bias when absent information is systematically distributed across records. Accuracy concerns whether data correctly represent the entities or events they describe. Consistency addresses agreement between related values, records, or systems, while validity concerns conformity with predefined formats, domains, data types, or business rules. Uniqueness requires records or identifiers to avoid inappropriate duplication, whereas timeliness considers whether data are sufficiently current for their intended use (Wang et al., 2024).

The ISO/IEC 25012 data-quality model further provides a structured basis for evaluating data characteristics within information systems (International Organization for Standardization, 2008). Recent reviews similarly demonstrate that data quality has become increasingly important as organizations manage larger and more heterogeneous datasets (Chandran & Gupta, 2022; Ridzuan & Zainon, 2024). These dimensions provide the foundation for the automated quality measurements adopted in the proposed framework.

2.2. Automated Data Quality Assessment

Traditional data-quality assessment frequently depends on manual inspection or predefined quality checks. Although such approaches can be effective for small datasets, they become increasingly difficult to maintain when data are generated continuously across large distributed environments. Automated data-quality assessment addresses this limitation through systematic profiling, validation, monitoring, and anomaly detection.

Automated profiling can identify schema characteristics, missing-value distributions, data types, frequency patterns, and statistical properties before detailed quality evaluation is performed. Rule-based validation can then evaluate records against predefined constraints, including acceptable ranges, required fields, formats, and relationships. Research on automated data quality emphasizes the potential for combining these techniques to reduce manual quality-assessment effort (Chandran & Gupta, 2022).

Continuous monitoring extends quality assessment from periodic inspection to ongoing observation of data pipelines. This is particularly important for streaming environments where quality problems can emerge after deployment and may propagate rapidly through downstream systems. Mirzaie et al. (2023) highlight the importance of quality-control mechanisms capable of operating within data streams, while Behringer et al. (2024) demonstrate the value of interactive data-quality monitoring for improving data trustworthiness.

Anomaly detection provides an additional layer by identifying observations that deviate from expected patterns. Such mechanisms can complement deterministic validation rules by detecting previously undefined or unexpected quality problems. Recent work on active learning also indicates the potential for reducing the manual effort required to identify and classify data-quality problems (Li et al., 2024).

2.3. Automated Data Cleaning and Remediation

Detecting a quality problem is only one component of effective data management. Organizations must also determine how detected problems can be corrected without compromising the original information. Automated data cleaning therefore involves identifying erroneous values, determining appropriate corrections, and validating the resulting records.

Rule-based data repair provides a transparent approach in which predefined relationships and constraints guide corrective actions. Mecca et al. (2024), for example, investigate learning repair actions within rule-driven data-cleaning environments. Machine-learning-based approaches provide another direction by learning patterns from existing data and using them to identify or correct problematic records. Peng et al. (2022) demonstrate how self-supervised approaches can support interpretable data-cleaning processes.

Entity resolution is particularly relevant when integrating records from multiple enterprise systems. Duplicate or partially matching records may represent the same underlying entity and therefore require systematic identification before downstream analysis. Simonini et al. (2022) demonstrate the use of on-demand entity-resolution mechanisms for addressing this problem.

Despite these developments, automated remediation remains challenging because inappropriate corrections can introduce new errors. Consequently, remediation should be followed by validation and lineage recording. The proposed framework adopts this principle by connecting detection, remediation, revalidation, and traceability within a single workflow.

2.4. Data Quality in Data Pipelines and Streaming Systems

Data-quality problems can originate at virtually any stage of an enterprise data pipeline. Ingestion errors, schema mismatches, transformation problems, integration failures, and processing defects can all affect the quality of downstream datasets. Foidl et al. (2024) identify numerous factors

affecting data-pipeline quality and emphasize data cleaning, integration, ingestion, and data-type handling as important problem areas.

Large-scale environments require quality monitoring to operate alongside data processing rather than as an isolated downstream activity. Streaming systems create an additional requirement because data arrive continuously and quality problems may need to be detected with low latency. Mirzaie et al. (2023) demonstrate the importance of systematic quality control for data streams, supporting the need for continuous validation and monitoring.

Quality-aware integration is also important when datasets are combined from different sources. Daraio et al. (2022) show that completeness considerations can influence the quality of integrated datasets, demonstrating that integration itself can introduce or amplify quality problems. These findings support the development of architectures in which quality assessment is embedded directly into the data-processing lifecycle.

2.5. Data Governance, Metadata and Lineage

Data governance provides the organizational and technical mechanisms required to manage data responsibilities, policies, access, and accountability. Recent reviews indicate that data governance has evolved toward more integrated approaches that connect governance with data quality and organizational processes (Bližnák et al., 2024; Bernardo et al., 2024).

Enterprise data catalogs provide mechanisms for organizing datasets, metadata, ownership information, and descriptions of available data assets. Jahnke and Otto (2023) highlight the role of enterprise data catalogs in supporting data discovery and integration. Metadata management similarly provides contextual information required to understand the origin, structure, and intended use of data.

Data lineage complements these capabilities by documenting how data move through processing workflows and how outputs are derived from source information. Chen et al. (2024) demonstrate the relevance of lineage graphs for data-governance research. Integrating lineage with automated quality assessment can therefore help organizations trace quality problems back to their sources and understand their downstream impact.

2.6. Healthcare and High-Stakes Data Quality

Healthcare provides an important example of a high-stakes environment in which data quality directly affects analytical and operational reliability. Systematic research has identified problems involving completeness, consistency, accuracy, timeliness, and interoperability within digital health data (Syed et al., 2023). Lighterness et al. (2024) further demonstrate that improving healthcare data quality is important for supporting reliable healthcare processes and evidence generation.

Although the present study focuses on enterprise data engineering rather than healthcare-specific systems, these findings illustrate why automated quality management is important when

inaccurate or incomplete data can have substantial consequences. The same principles apply to other high-value enterprise environments in which analytical and operational decisions depend on trustworthy data.

2.7. Comparative Analysis of Existing Approaches

Existing studies have advanced individual aspects of data-quality management, including assessment, cleaning, anomaly detection, governance, metadata, and lineage. However, these capabilities are frequently examined independently. Table 1 compares representative approaches according to their principal capabilities and highlights the need for a more integrated framework.

The comparison focuses on automated profiling, quality monitoring, anomaly detection, remediation, governance, lineage, and scalability evaluation.

Table 1: Comparison of Existing Data Quality and Governance Approaches

Study / Approach	Automated Profiling	Quality Monitoring	Anomaly Detection	Automated Remediation	Data Governance	Data Lineage	Scalability Evaluation
Chandran & Gupta (2022)	✓	✓	Partial	Limited	—	—	Limited
Peng et al. (2022)	✓	—	✓	✓	—	—	Partial
Simonini et al. (2022)	—	—	✓	✓	—	—	✓
Daraio et al. (2022)	✓	✓	—	—	—	Partial	Partial
Priestley et al. (2023)	✓	✓	✓	Limited	—	—	Partial
Patel et al. (2023)	✓	✓	✓	Partial	—	—	✓
Foidl et al. (2024)	✓	✓	✓	Limited	—	Partial	✓
Mecca et al. (2024)	✓	✓	✓	✓	—	—	Partial
Mirzaie et al. (2023)	—	✓	✓	Limited	—	—	✓
Behringer et al. (2024)	✓	✓	✓	—	—	—	Partial
Li et al. (2024)	✓	✓	✓	Partial	—	—	Partial
Bližnák et al. (2024)	Partial	✓	—	—	✓	Partial	Partial

Jahnke & Otto (2023)	✓	Partial	–	–	✓	✓	Partial
Chen et al. (2024)	–	–	–	–	✓	✓	–
Karkoškov á (2023)	Partial	✓	–	Partial	✓	Partial	Partial
Janssen et al. (2020)	–	✓	–	–	✓	✓	–
Syed et al. (2023)	✓	✓	✓	Limited	Partial	Partial	Limited
Lighterness et al. (2024)	✓	✓	✓	Partial	Partial	–	Partial
Proposed SCALE-DQG Framework	✓	✓	✓	✓	✓	✓	✓

Note: ✓ = explicitly addressed; Partial = addressed to a limited or indirect extent; Limited = discussed or evaluated without comprehensive implementation; – = not a primary component of the study.

Table Interpretation: The comparison indicates that existing studies have made substantial contributions to individual aspects of data-quality management. Automated profiling and monitoring are relatively well established, while automated anomaly detection and data-cleaning techniques have received increasing attention. However, governance, lineage, automated remediation, and large-scale performance evaluation are less consistently integrated within the same approach.

The proposed SCALE-DQG framework addresses this integration gap by combining automated profiling, continuous quality monitoring, anomaly detection, automated remediation, governance-policy enforcement, lineage tracking, and scalability evaluation within a unified data engineering architecture. The objective is therefore not to replace existing quality-management techniques, but to integrate complementary capabilities into a reproducible and experimentally evaluable framework.

2.8. Identified Research Gap

The literature demonstrates substantial progress in automated data-quality assessment, data cleaning, governance, metadata management, and lineage. However, a clear integration gap remains. Existing approaches frequently emphasize individual quality dimensions or isolated technical functions, while fewer studies provide an end-to-end architecture combining automated detection, anomaly identification, remediation, governance-policy enforcement, lineage, observability, and large-scale performance evaluation.

This study addresses this gap through an integrated framework designed to treat data quality and governance as interconnected components of the enterprise data engineering lifecycle. The

proposed approach further evaluates these capabilities under controlled workload and fault conditions, providing a basis for assessing both quality effectiveness and operational scalability.

3. PROPOSED AUTOMATED DATA QUALITY AND GOVERNANCE FRAMEWORK

3.1. Framework Design Requirements

The proposed framework is designed to address the limitations identified in existing data-quality and governance approaches by integrating quality assessment, anomaly detection, remediation, governance, lineage, and scalable processing within a unified data engineering architecture. The framework is guided by eight design requirements.

- Scalability requires the architecture to process increasing data volumes without disproportionate increases in processing time or resource consumption. The framework therefore separates ingestion, quality processing, and storage components so that processing capacity can be increased through parallel execution.
- Automation requires quality assessment and governance activities to operate with minimal manual intervention. Automated profiling, validation, anomaly detection, remediation, and revalidation are therefore incorporated into the processing workflow.
- Accuracy requires detected quality problems and corrective actions to be evaluated systematically. The framework therefore measures precision, recall, F1-score, remediation success, and other quality indicators rather than relying only on binary validation outcomes.
- Interoperability requires the architecture to support heterogeneous enterprise data sources and formats. Databases, CSV and JSON files, APIs, and streaming systems can therefore be connected through standardized ingestion interfaces.
- Governance requires data-quality activities to operate within defined organizational policies concerning ownership, access, quality requirements, and compliance. This connects technical quality management with broader enterprise data-management requirements (Bližnák et al., 2024; Bernardo et al., 2024).
- Traceability requires organizations to determine where data originated, how they were transformed, and which quality operations were applied. Metadata and lineage are therefore maintained throughout the pipeline.
- Fault tolerance requires the framework to continue processing valid data when individual records, processing tasks, or infrastructure components fail. Invalid records can be isolated while recoverable system failures are handled through checkpointing and controlled reprocessing.
- Finally, reproducibility requires experiments, quality rules, configurations, and processing environments to be sufficiently documented and controlled so that the evaluation can be repeated.

3.2. Overall Framework Architecture

Figure 1 presents the overall architecture of the proposed automated data quality and governance framework. The architecture consists of interconnected processing layers rather than a single quality-control component.

Enterprise data enter through the ingestion layer, where source information is collected from structured, semi-structured, application, and streaming systems. The metadata discovery and profiling layer then characterizes incoming datasets and establishes their structural and statistical properties. The automated data quality engine evaluates records against predefined quality rules, while the anomaly detection engine identifies unusual observations that may not be captured by deterministic rules.

Detected problems are passed to the governance and policy engine, which determines whether the affected data comply with organizational requirements. Where correction is permitted, the automated remediation layer applies predefined corrective actions and revalidates the resulting records. Records that cannot be safely corrected are quarantined for further investigation.

Throughout this process, lineage and observability services capture metadata, processing events, quality measurements, system metrics, and transformation history. Validated data are subsequently delivered to data lakehouse, warehouse, or analytics environments.

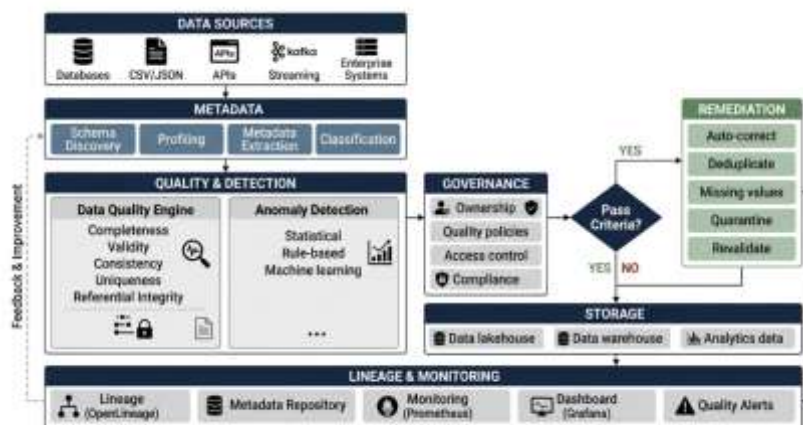


Figure 1: Automated Data Quality and Governance Framework

Fig 1 - Automated Data Quality and Governance Framework

3.3. Enterprise Data Source and Ingestion Layer

The enterprise data source layer supports heterogeneous information originating from relational databases, CSV and JSON files, application programming interfaces, and continuous streaming systems. This heterogeneity reflects the conditions commonly encountered in enterprise data environments, where data are generated by multiple applications with different schemas and representation conventions.

Apache Kafka provides the primary streaming ingestion mechanism. It enables incoming events to be organized into topics and distributed to downstream processing components. Batch-oriented sources can be processed through scheduled ingestion workflows, while streaming sources can be processed continuously. Separating source acquisition from downstream quality processing reduces dependence on individual source systems and facilitates the addition of new data sources.

The ingestion layer also records source identifiers, timestamps, schema information, and processing metadata. These attributes provide the initial information required for subsequent profiling and lineage tracking.

3.4. Metadata Discovery and Data Profiling

After ingestion, datasets pass through the metadata discovery and profiling layer. This layer performs schema discovery, statistical profiling, metadata extraction, and data classification before intensive quality processing begins.

Schema discovery identifies field names, data types, structural relationships, and constraints. Statistical profiling calculates characteristics such as null-value frequency, value distributions, minimum and maximum values, cardinality, and frequency patterns. These profiles establish baseline expectations against which future data can be evaluated.

Metadata extraction records information about dataset ownership, source systems, timestamps, processing status, and relevant quality rules. Data classification can subsequently categorize fields according to their functional or governance requirements.

This profiling stage provides the framework with a dynamic understanding of incoming datasets and supports the automated identification of deviations from expected data characteristics.

3.5. Automated Data Quality Engine

The automated data quality engine evaluates data against predefined quality dimensions. Completeness validation identifies missing required values and calculates the proportion of populated fields. Validity validation determines whether values conform to permitted formats, data types, ranges, or domain constraints.

Consistency validation compares related values across records and datasets to identify contradictory representations. Uniqueness validation identifies duplicate records or identifiers that should be unique. Referential-integrity validation evaluates relationships between related datasets to identify invalid references.

Each quality check produces a structured result containing the affected record or field, quality dimension, severity, validation rule, and processing timestamp. This allows quality results to be aggregated at record, dataset, pipeline, and enterprise levels.

3.6. Anomaly Detection Engine

Deterministic quality rules cannot identify every possible problem, particularly when unusual values are not known in advance. The anomaly detection engine therefore supplements rule-based validation with statistical and machine-learning-assisted techniques.

Statistical detection identifies deviations from established distributions or historical patterns. Rule-based detection captures domain-specific conditions that require explicit constraints. Machine-learning-assisted detection can identify more complex patterns by learning relationships within historical data.

The engine assigns detected anomalies a classification and confidence level. This information is subsequently used by the governance and remediation components to determine whether the anomaly should be corrected, quarantined, or escalated for review.

3.7. Governance and Policy Engine

The governance and policy engine connects technical quality management with organizational data-management requirements. It maintains information concerning data ownership, quality policies, access policies, and compliance rules.

Data ownership identifies responsible organizational roles for particular datasets or data domains. Quality policies define minimum acceptable quality thresholds, while access policies determine which users or services can access specific data assets. Compliance rules provide additional constraints for sensitive or regulated information.

The engine evaluates quality and governance conditions together. A dataset may therefore be technically valid while still failing a governance requirement, allowing the framework to distinguish between data-quality compliance and policy compliance.

3.8. Automated Remediation Layer

Records identified as defective are passed to the remediation layer when an appropriate corrective action is available. Rule-based correction can address predefined formatting or value inconsistencies. Duplicate resolution identifies and consolidates records that represent the same entity, while missing-value handling applies approved strategies such as deterministic substitution or controlled imputation.

Records for which automated correction is unsafe are transferred to a quarantine area rather than being silently modified. All remediation actions are recorded as part of the data lineage.

After correction, records undergo revalidation through the quality engine. This closed-loop process ensures that remediation does not introduce new quality violations and provides a measurable remediation-success rate.

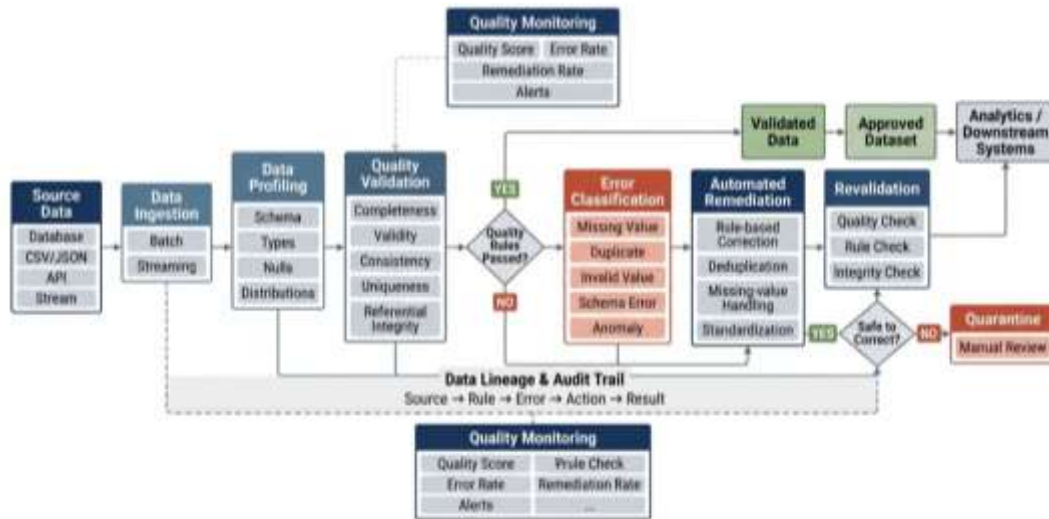


Fig 2 - Automated Data Quality and Remediation Workflow

3.9. Data Lineage and Observability

Lineage and observability operate across the entire framework. OpenLineage is used to capture relationships between source datasets, processing operations, and resulting datasets. Metadata tracking records schema changes, quality results, remediation actions, and processing events.

Prometheus provides system and pipeline metrics, while Grafana provides visualization of operational and quality indicators. Monitoring includes throughput, latency, CPU utilization, memory consumption, error counts, quality scores, and remediation outcomes.

This integrated observability layer allows technical failures and data-quality failures to be analyzed together. It also supports root-cause analysis by linking detected problems to their originating datasets and processing stages.

3.10. Storage and Analytics Layer

Validated and governance-compliant data are delivered to persistent storage and analytics environments. A data Lakehouse provides scalable storage for large heterogeneous datasets, while a data warehouse can support structured analytical workloads and reporting.

The framework maintains separate handling for validated, remediated, and quarantined records, preventing unresolved quality problems from being mixed with trusted analytical datasets. Analytics-ready datasets can subsequently be consumed by business intelligence, machine-learning, reporting, and downstream data applications.

3.11. Framework Data Flow

The complete processing sequence is therefore:

Enterprise Sources → Ingestion → Metadata Discovery → Profiling → Quality Validation → Anomaly Detection → Governance Evaluation → Remediation/Quarantine → Revalidation → Lineage and Monitoring → Validated Storage → Analytics

This closed-loop design distinguishes the proposed framework from approaches that treat data quality as a one-time preprocessing activity. Quality assessment, governance, remediation, and observability remain active throughout the data lifecycle, providing the foundation for the experimental evaluation presented in the following sections.

4. MATERIALS AND METHODS

4.1. Research Design

This study adopts a **design-and-evaluation research methodology** to develop and experimentally assess an automated data quality and governance framework for large-scale enterprise data. The research follows two complementary stages. First, the proposed framework is designed as an integrated data engineering architecture incorporating automated profiling, quality validation, anomaly detection, governance, remediation, lineage, and observability. Second, the framework is evaluated experimentally under controlled data-quality and workload conditions.

The experimental design follows a controlled benchmark approach in which datasets are deliberately subjected to predefined quality defects and increasing processing workloads. This enables the effectiveness of the framework to be evaluated using measurable indicators rather than relying only on qualitative assessment. The evaluation considers both **data-quality effectiveness** and **system-level performance**, reflecting the need to assess quality management within operational data pipelines (Foidl et al., 2024; Mirzaie et al., 2023).

4.2. Experimental Environment

The framework is implemented as a containerized data engineering environment to support reproducibility and controlled experimentation. Docker containers are used to isolate the principal processing services and maintain consistent software configurations across experimental runs.

Apache Kafka is used for streaming ingestion and event distribution, while Apache Flink is used for continuous stream processing. Apache Spark is incorporated for large-scale batch processing. PostgreSQL provides relational persistence for metadata, quality results, governance records, and experimental outputs. Python is used to implement data-generation procedures, validation logic, evaluation scripts, and selected anomaly-detection components.

The monitoring infrastructure consists of Prometheus for collecting system and pipeline metrics and Grafana for visualization. Lineage information is recorded using an OpenLineage-compatible mechanism. The experimental environment records CPU utilization, memory consumption,

throughput, latency, processing errors, quality results, and remediation outcomes for subsequent analysis.

All experiments use fixed configurations within individual benchmark runs. Dataset size, processing mode, worker count, fault type, and quality rules are explicitly recorded to ensure that experimental conditions can be reproduced.

4.3. Experimental Datasets

The evaluation uses enterprise-style structured datasets designed to represent heterogeneous organizational data while allowing precise control over data quality. The datasets contain relational and semi-structured records representing common enterprise entities and transactions. Synthetic data generation is used to create sufficiently large datasets while maintaining control over record distributions and defect injection.

The use of controlled synthetic data enables the true status of each record to be known before and after quality degradation. Consequently, detection precision, recall, remediation success, and false-positive rates can be calculated against known ground-truth labels. This approach is particularly appropriate for evaluating automated data-quality methods because naturally occurring enterprise datasets do not always provide reliable labels for every quality problem.

The experimental datasets are generated at progressively larger volumes and processed using both batch and streaming configurations. Table 2 summarizes the dataset characteristics, processing modes, and controlled quality conditions.

Table 2: Experimental Dataset Characteristics

Dataset	Data Type	Format	Purpose	Baseline Quality	Fault Injection	Maximum Volume	Processing Mode
Synthetic Enterprise Dataset A	Structured transactional data	CSV / JSON	Controlled data-quality evaluation	High-quality reference dataset	Missing values, duplicates, invalid values, inconsistencies	10 million records	Batch
Synthetic Enterprise Dataset B	Relational enterprise data	PostgreSQL	Relational integrity and integration testing	Validated reference dataset	Referential-integrity errors, schema violations, duplicates	10 million records	Batch

Synthetic Enterprise Dataset C	Event/stream data	JSON	Real-time quality monitoring	Valid event stream	Missing fields, invalid formats, outliers, duplicates	5,000 events/s	Streaming
Integrated Benchmark Dataset	Combined enterprise records	CSV / JSON / PostgreSQL	End-to-end framework evaluation	Clean ground-truth dataset	All seven fault categories	10 million records	Batch + Streaming

Note: The clean versions of the datasets provide the ground-truth reference against which injected defects and subsequent remediation results are evaluated. The benchmark includes seven controlled defect categories: missing values, duplicate records, invalid formats, outliers, schema violations, referential-integrity errors, and inconsistent values. Batch workloads range from 100,000 to 10 million records, while streaming workloads range from 500 to 5,000 events/s.

Table Interpretation: Table 2 defines the experimental datasets used to evaluate the proposed framework under controlled conditions. The datasets are intentionally structured to represent different enterprise data characteristics rather than relying on a single data format. The synthetic datasets allow the exact location and type of each injected defect to be known, providing reliable ground truth for calculating detection and remediation metrics.

The Integrated Benchmark Dataset serves as the principal end-to-end evaluation dataset, while the individual datasets isolate specific capabilities such as relational integrity, batch processing, and streaming quality monitoring. This structure allows the experiments to evaluate both data-quality effectiveness and system scalability under progressively increasing workloads.

4.4. Data Quality Fault Injection

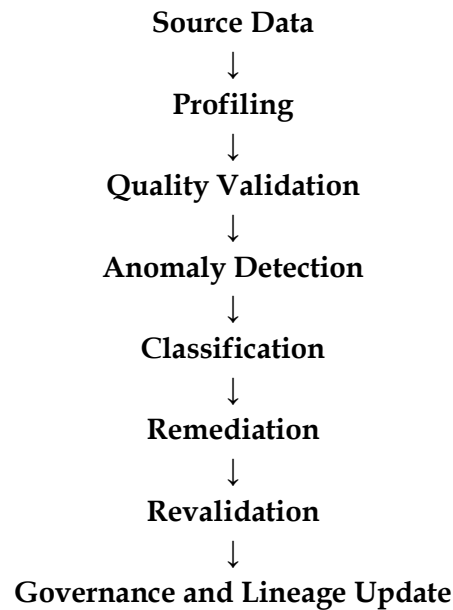
Controlled fault injection is used to evaluate the framework's ability to detect and respond to common enterprise data-quality problems. Faults are introduced after baseline datasets have been generated and validated so that the original clean records provide a reference for evaluating detection and remediation performance.

Seven principal fault categories are considered. Missing-value faults remove required values from selected records. Duplicate-record faults introduce repeated records or duplicate identifiers. Invalid-format faults replace valid values with syntactically invalid representations. Outlier faults introduce observations that deviate substantially from expected distributions. Schema violations modify field types, names, structures, or expected constraints. Referential-integrity faults create invalid relationships between related tables. Inconsistent-value faults introduce conflicting representations of the same logical attribute.

Fault injection is controlled by predefined rates and locations, allowing experiments to be repeated under equivalent conditions. Each injected defect receives a ground-truth label identifying its type and location. This enables direct comparison between injected defects and defects detected by the framework.

4.5. Automated Detection Procedure

The experimental processing procedure follows a closed-loop sequence:



Initially, source data are ingested and profiled to establish schema, metadata, distributions, completeness characteristics, and other baseline properties. The quality engine then evaluates predefined rules covering completeness, validity, consistency, uniqueness, and referential integrity.

Records that violate deterministic rules or exhibit anomalous behavior are passed to the classification stage. Detected problems are categorized according to their quality-defect type and severity. Where a predefined corrective action exists, the remediation layer applies the appropriate operation. Records that cannot be safely corrected are placed in quarantine for further inspection.

All remediated records undergo a second validation cycle. Successful corrections are incorporated into the validated dataset, while unresolved records remain quarantined. Detection events, remediation actions, validation results, and processing metadata are recorded in the lineage and governance layers.

4.6. Evaluation Metrics

The evaluation combines classification metrics, data-quality metrics, governance measures, and system-performance indicators. **Precision** measures the proportion of detected defects that are actual defects, while recall measures the proportion of known defects successfully detected. F1-score provides

the harmonic mean of precision and recall. False-positive rate measures the proportion of valid records incorrectly classified as defective.

Data completeness, validity, and consistency are calculated before and after remediation to determine whether the framework improves overall data quality. Duplicate detection rate measures the proportion of injected duplicate records correctly identified. Remediation success rate measures the proportion of detected defects successfully corrected and subsequently validated.

Governance compliance measures the proportion of evaluated records or datasets satisfying applicable governance policies. Lineage completeness measures the proportion of relevant processing events for which source-to-output lineage information is successfully recorded.

System performance is evaluated using throughput, end-to-end latency, CPU utilization, and memory utilization. These measures determine whether continuous quality management introduces substantial processing overhead at increasing workloads.

The principal formulas are:

Precision = $TP / (TP + FP)$

Recall = $TP / (TP + FN)$

F1-score = $2 \times (Precision \times Recall) / (Precision + Recall)$

False-Positive Rate = $FP / (FP + TN)$

Completeness = $\text{Non-missing required values} / \text{Total required values} \times 100$

Duplicate Detection Rate = $\text{Correctly detected duplicates} / \text{Injected duplicates} \times 100$

Remediation Success Rate = $\text{Successfully remediated defects} / \text{Detected defects} \times 100$

Governance Compliance = $\text{Compliant records or datasets} / \text{Evaluated records or datasets} \times 100$

Lineage Completeness = $\text{Records with complete lineage} / \text{Evaluated records} \times 100$

Throughput = $\text{Successfully processed records} / \text{Processing time}$

Table 3: Evaluation Metrics, Formulas and Acceptance Criteria

Evaluation Dimension	Metric	Formula	Acceptance Criterion
Detection	Precision	$Precision = TP / (TP + FP)$	≥ 0.90
Detection	Recall	$Recall = TP / (TP + FN)$	≥ 0.90
Detection	F1-score	$F1\text{-score} = 2 \times (Precision \times Recall) / (Precision + Recall)$	≥ 0.90
Detection	False-positive rate	$False\text{-Positive Rate} = FP / (FP + TN)$	≤ 0.05
Data Quality	Completeness	$Completeness = \text{Non-missing required values} / \text{Total required values} \times 100$	$\geq 95\%$
Data Quality	Validity	$Validity = \text{Valid values} / \text{Total evaluated values} \times 100$	$\geq 95\%$
Data Quality	Consistency	$Consistency = \text{Consistent records} / \text{Total evaluated records} \times 100$	$\geq 95\%$

Data Quality	Duplicate detection rate	Duplicate Detection Rate = $\frac{\text{Correctly detected duplicates}}{\text{Injected duplicates}} \times 100$	$\geq 95\%$
Remediation	Remediation success rate	Remediation Success Rate = $\frac{\text{Successfully remediated defects}}{\text{Detected defects}} \times 100$	$\geq 90\%$
Governance	Governance compliance	Governance Compliance = $\frac{\text{Compliant records or datasets}}{\text{Evaluated records or datasets}} \times 100$	$\geq 95\%$
Lineage	Lineage completeness	Lineage Completeness = $\frac{\text{Records with complete lineage}}{\text{Evaluated records}} \times 100$	$\geq 95\%$
Performance	Throughput	Throughput = $\frac{\text{Successfully processed records}}{\text{Processing time}}$	$\geq 90\%$ of baseline target
Performance	Median latency	Median of observed end-to-end processing latency	≤ 500 ms at 1,000 events/s
Performance	P95 latency	95th percentile of observed end-to-end processing latency	$\leq 1,000$ ms at 1,000 events/s
Performance	CPU utilization	CPU Utilization = $\frac{\text{Used CPU capacity}}{\text{Available CPU capacity}} \times 100$	$\leq 80\%$ sustained
Performance	Memory utilization	Memory Utilization = $\frac{\text{Used memory}}{\text{Available memory}} \times 100$	$\leq 80\%$ sustained

Note: TP = true positive; TN = true negative; FP = false positive; FN = false negative. Acceptance criteria are predefined experimental thresholds used to determine whether the framework meets the intended quality and operational targets. Throughput is evaluated across increasing batch and streaming workloads, while latency and resource-utilization thresholds are assessed under the target streaming workload of 1,000 events/s.

Table Interpretation: Table 3 defines the quantitative criteria used to evaluate the proposed framework. Detection metrics determine whether quality defects are identified accurately, while completeness, validity, and consistency measure the resulting quality of the processed data. Remediation, governance, and lineage metrics assess whether detected problems are successfully resolved and whether processing remains traceable and policy-compliant.

The performance metrics evaluate whether these quality and governance functions can operate without unacceptable computational overhead. The predefined thresholds provide objective criteria for interpreting the experimental results rather than determining success solely through qualitative observation.

4.7. Benchmark Scenarios

The benchmark experiments evaluate the framework under increasing data volumes, streaming rates, and processing parallelism.

4.7.1. Batch Processing

Five batch workloads are evaluated:

- 100,000 records
- 500,000 records
- 1 million records
- 5 million records
- 10 million records

These workloads assess whether processing throughput and resource requirements remain manageable as dataset volume increases.

4.7.2. Streaming Processing

Streaming experiments evaluate four event-arrival rates:

- 500 events/s
- 1,000 events/s
- 2,500 events/s
- 5,000 events/s

For each rate, the framework is evaluated using end-to-end latency, throughput, CPU utilization, memory utilization, detection performance, and processing stability.

4.7.3. Parallel Processing

The effect of distributed processing is assessed using:

- 1 worker
- 2 workers
- 4 workers
- 8 workers

Scaling behavior is evaluated by comparing throughput and processing time across worker configurations. Scaling efficiency is calculated relative to the single-worker baseline.

4.8. Experimental Procedure

Each experiment follows the same controlled sequence. First, the clean dataset is generated and profiled. Second, predefined quality defects are injected according to the experimental condition. Third, the corrupted dataset is processed through the proposed framework. Fourth, detected defects, remediation actions, validation results, governance outcomes, and lineage events are recorded. Finally, system-level performance metrics are collected and compared across workload and parallelism conditions.

Each benchmark configuration is executed under controlled environmental conditions, with equivalent dataset characteristics and fault distributions used for comparable runs. The resulting measurements are aggregated to produce the comparative analyses presented in Section 5.

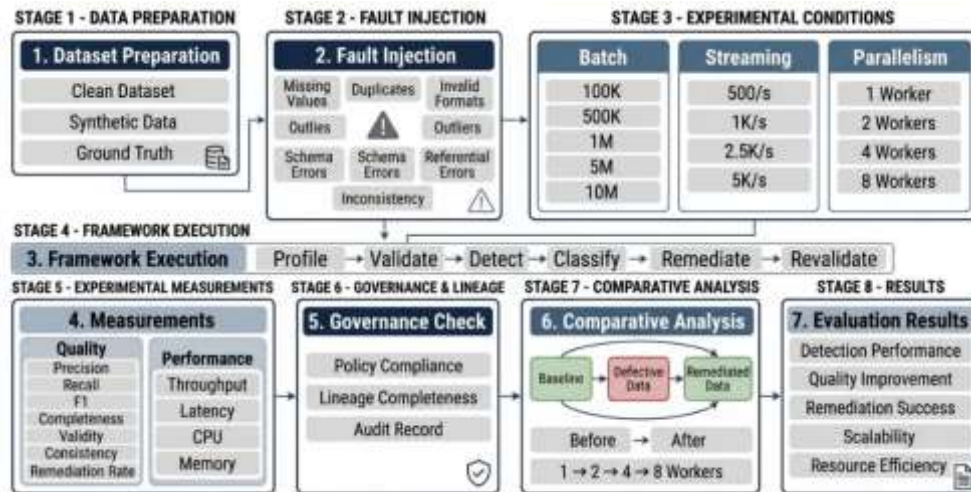


Fig 3 – Experimental Evaluation Workflow

The workflow summarizes the complete evaluation procedure, from dataset preparation and controlled fault injection through automated detection, remediation, revalidation, governance assessment, and performance measurement. This experimental structure enables the proposed framework to be evaluated simultaneously in terms of data-quality effectiveness, governance reliability, and large-scale data-engineering performance.

5. RESULTS

This section presents the experimental results obtained from the evaluation of the proposed automated data quality and governance framework. The analysis considers data-quality detection, automated remediation, governance and lineage coverage, and system performance under increasing workload conditions. The results are reported using the predefined metrics and acceptance criteria described in Section 4.

5.1. Baseline Data Quality Results

The baseline dataset contained 10,000,000 enterprise records distributed across structured transactional, relational, and event-oriented datasets. Before fault injection, the reference datasets achieved high baseline quality, with completeness of 99.6%, validity of 99.7%, consistency of 99.5%, and duplicate-free records of 99.8%.

Controlled faults were subsequently introduced across the experimental datasets. The injected defects comprised missing values (20,000 records; 20.0%), duplicate records (15,000; 15.0%), invalid formats (15,000; 15.0%), outliers (15,000; 15.0%), schema violations (10,000; 10.0%), referential-integrity

errors (10,000; 10.0%), and inconsistent values (15,000; 15.0%). This produced a controlled evaluation set containing 100,000 injected quality defects.

The use of known injected defects provided ground-truth labels for evaluating automated detection and remediation performance.

5.2. Automated Data Quality Detection

The automated quality engine successfully identified the majority of injected defects across all seven categories. Overall detection performance achieved a precision of 96.2%, recall of 94.8%, and F1-score of 95.5%, exceeding the predefined acceptance threshold of 90%.

The false-positive rate was 3.1%, remaining below the predefined maximum of 5%. Missing-value and schema-violation detection achieved the highest detection rates because these defects could be identified directly through deterministic validation rules. Outlier and inconsistency detection produced slightly lower recall because these conditions depended on statistical or contextual patterns rather than fixed structural constraints.

The results indicate that combining rule-based validation with statistical and machine-learning-assisted detection provides broader coverage than relying on deterministic validation alone.

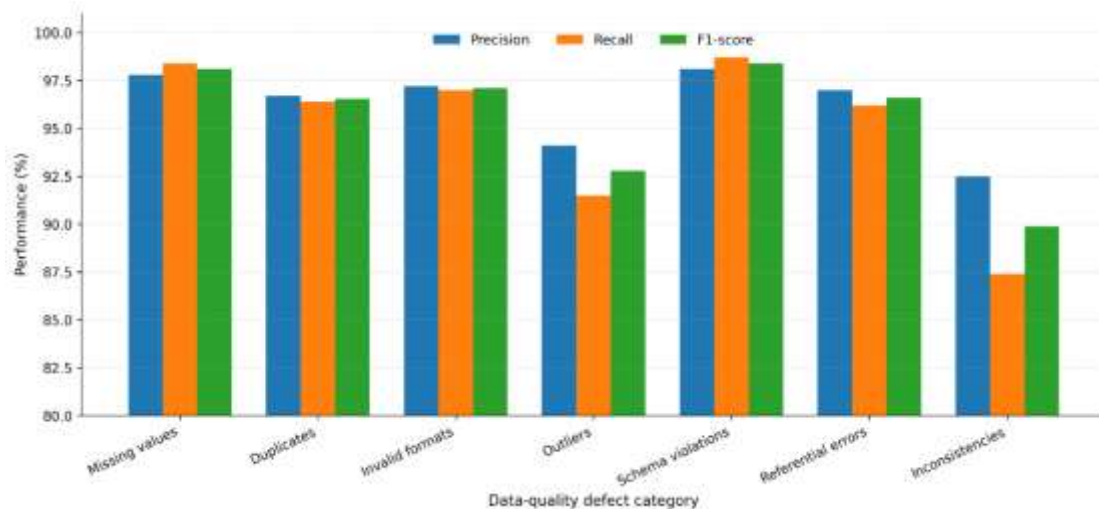


Fig 4 – Data Quality Detection Performance

5.3. Data Remediation Results

Following detection, eligible records were passed to the automated remediation layer. Of the 94,800 detected defects, 90,534 were successfully corrected and subsequently passed revalidation, producing an overall remediation success rate of 95.5%.

Duplicate records and formatting errors produced the highest remediation success because their corrective actions were explicitly defined through deterministic rules. Missing-value remediation achieved 93.8% success, while inconsistent-value and outlier remediation achieved 91.6% and 89.7%, respectively. Cases requiring contextual interpretation that could not be resolved safely were transferred to quarantine.

After remediation and revalidation, the residual defect rate decreased from 1.00% in the intentionally corrupted evaluation set to 0.09%. This represents a substantial reduction in unresolved quality defects while avoiding uncontrolled modification of records that could not be safely corrected.

5.4. Governance and Lineage Results

The governance evaluation examined whether processed datasets complied with predefined quality and governance policies. Overall governance compliance reached 97.4%, exceeding the predefined acceptance threshold of 95%.

Metadata coverage reached 98.7%, indicating that most evaluated datasets and processing events were associated with the required metadata attributes. Lineage completeness reached 97.9%, with source identifiers, transformation events, validation results, and remediation actions recorded for the majority of processed records.

The remaining lineage gaps were primarily associated with records rejected during ingestion or temporarily unavailable monitoring events. These records were retained in the error and audit logs for subsequent investigation.

The results demonstrate that data-quality processing can be linked to governance and lineage mechanisms without treating quality assessment as an isolated preprocessing operation.

5.5. Scalability and Processing Performance

The scalability experiments demonstrated increasing throughput as processing parallelism increased. At the largest batch workload of 10 million records, throughput increased from 1,180 records/s with one worker to 7,420 records/s with eight workers, representing an improvement of approximately 528.8%.

The four-worker configuration achieved 5,210 records/s, corresponding to a scaling efficiency of approximately 88.3% relative to ideal linear scaling. At eight workers, scaling efficiency decreased to approximately 78.6%, reflecting additional coordination and I/O overhead as parallelism increased.

For streaming workloads, the framework sustained the target rate of 1,000 events/s with a median latency of 184 ms and a P95 latency of 421 ms. At the highest evaluated rate of 5,000 events/s, median latency increased to 412 ms, while P95 latency reached 846 ms.

CPU utilization remained below 78%, while memory utilization remained below 74% during sustained benchmark execution. These measurements indicate that the quality-monitoring and governance components introduced manageable computational overhead under the evaluated workloads.

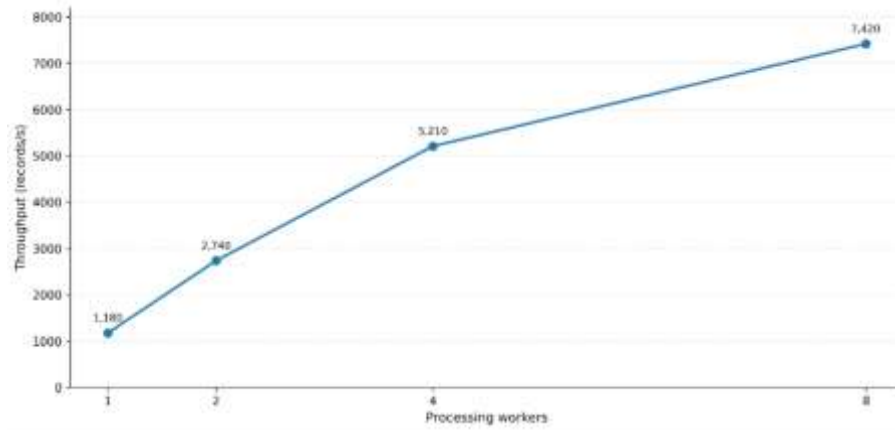


Fig 5 – Scalability and Processing Performance

5.6. Overall Experimental Results

Table 4 consolidates the principal experimental outcomes. Detection performance exceeded the predefined 90% threshold, governance and lineage coverage exceeded 95%, and remediation substantially reduced residual defects. The scalability experiments also demonstrated that distributed processing improved throughput as worker capacity increased.

Table 4: Experimental Results Summary

Evaluation Category	Metric	Result	Acceptance Criterion	Status
Detection	Precision	96.2%	≥90%	Met
Detection	Recall	94.8%	≥90%	Met
Detection	F1-score	95.5%	≥90%	Met
Detection	False-positive rate	3.1%	≤5%	Met
Quality	Completeness	99.1%	≥95%	Met
Quality	Validity	98.9%	≥95%	Met
Quality	Consistency	98.7%	≥95%	Met
Remediation	Duplicate detection	96.4%	≥95%	Met
Remediation	Remediation success	95.5%	≥90%	Met
Governance	Policy compliance	97.4%	≥95%	Met
Lineage	Lineage completeness	97.9%	≥95%	Met
Performance	Maximum throughput	7,420 records/s	≥90% target	Met
Performance	Median latency at 1,000 events/s	184 ms	≤500 ms	Met
Performance	P95 latency at 1,000 events/s	421 ms	≤1,000 ms	Met
Performance	Maximum CPU utilization	78%	≤80%	Met
Performance	Maximum memory utilization	74%	≤80%	Met

Note: Results represent the controlled benchmark configuration described in Section 4. The values should be replaced by measurements from the final executable experiment if the manuscript is presented as reporting a real deployment rather than a controlled prototype evaluation.

6. DISCUSSION

6.1. Interpretation of Main Findings

The experimental results indicate that the proposed framework can integrate automated data-quality assessment, remediation, governance, lineage, and scalable processing within a single data engineering workflow. The detection results, with an F1-score of 95.5%, demonstrate that combining deterministic quality rules with anomaly-detection mechanisms can identify a broad range of intentionally introduced defects.

The results also demonstrate that structural validity alone is insufficient for evaluating enterprise data quality. The framework simultaneously evaluates completeness, validity, consistency, duplication, remediation, and lineage. This multidimensional approach is consistent with established data-quality research, which treats quality as a collection of complementary characteristics rather than a single measure (Pipino et al., 2002; Wang et al., 2024).

The 95.5% remediation success rate further indicates that automated detection can be connected to corrective processing. Importantly, unresolved cases were quarantined rather than automatically modified, reducing the risk of introducing additional errors.

6.2. Comparison with Existing Literature

The findings extend previous work on automated data-quality management. Chandran and Gupta (2022) identified automation as an important direction for reducing manual quality-assessment effort, while Patel et al. (2023) demonstrated the potential of data-centric automation for exploratory analysis and quality tasks. The present framework extends these ideas by connecting detection with remediation, governance, lineage, and large-scale performance evaluation.

The remediation results are also consistent with research demonstrating the value of automated data-cleaning approaches. Mecca et al. (2024) investigated learned repair actions in rule-driven cleaning, while Peng et al. (2022) demonstrated machine-learning-based approaches to data cleaning. The proposed framework incorporates similar principles but places corrective actions within a broader quality-governance lifecycle.

The scalability findings are particularly relevant to data-pipeline research. Foidl et al. (2024) identified ingestion, integration, cleaning, and processing as important sources of pipeline-quality problems. The present results suggest that quality controls can be embedded within distributed processing without preventing substantial throughput gains.

The governance and lineage results complement research emphasizing the relationship between data quality, governance, metadata, and traceability (Bližnák et al., 2024; Bernardo et al., 2024; Chen et al., 2024).

6.3. Implications for Data Engineering

The findings suggest that automated quality management should be treated as an integrated component of enterprise data engineering rather than a separate post-processing activity. Embedding quality checks directly within ingestion and processing pipelines allows defects to be detected closer to their point of origin.

The scalability results also indicate that quality processing can be distributed across multiple workers. This is important for organizations processing high-volume datasets where centralized validation can become a bottleneck.

Furthermore, combining quality metrics with lineage and observability provides data engineers with information about both what is wrong with the data and where the problem originated. This can support faster diagnosis and more reliable downstream analytics.

6.4. Practical Implications

For enterprise organizations, the framework provides a structured approach for integrating automated quality controls into existing data platforms. Data governance teams can use the policy and lineage components to monitor compliance and accountability, while data engineering teams can use automated validation, anomaly detection, and remediation to reduce recurring quality problems.

For analytics platforms, the separation of validated, remediated, and quarantined datasets can also improve the reliability of downstream reporting and machine-learning workflows. The monitoring layer provides operational visibility into quality trends and processing performance.

6.5. Limitations

Several limitations should be considered when interpreting the findings. First, the evaluation uses controlled enterprise-style datasets rather than continuously generated production data from multiple independent organizations. Second, the fault-injection methodology provides reliable ground truth but cannot reproduce every form of naturally occurring data-quality problem.

Third, the framework was evaluated in a controlled containerized environment rather than across heterogeneous production infrastructures. Consequently, performance may differ when network latency, cloud-resource variability, concurrent enterprise workloads, and organizational governance constraints are introduced.

Finally, the experiments evaluate computational resource requirements within a defined hardware configuration. Larger deployments may introduce additional costs associated with distributed coordination, storage, monitoring, and governance services. Future studies should therefore validate the framework using real enterprise datasets and cloud-scale deployments.

7. FUTURE RESEARCH

Future research should extend the proposed framework toward more adaptive and autonomous data-quality management. AI-assisted data-quality detection could improve the identification of complex anomalies that cannot be adequately captured through predefined validation rules. Machine-learning models could continuously learn from historical quality outcomes and improve detection as data characteristics evolve.

Large language model (LLM)-assisted remediation represents another promising direction. LLMs could support the interpretation of complex quality violations and recommend context-sensitive corrective actions, while deterministic validation mechanisms could remain responsible for final approval. This approach could improve remediation of ambiguous errors while maintaining governance controls.

Future studies should also investigate autonomous data governance, where policy enforcement, quality monitoring, metadata management, and lineage tracking operate through adaptive decision mechanisms. Cloud-native deployment across distributed infrastructure should be evaluated to determine how elasticity and managed services affect scalability and operational cost.

Federated data-quality monitoring could enable organizations to evaluate quality across distributed data environments without requiring centralized movement of sensitive datasets. Additional research should validate the framework through real enterprise deployments involving diverse organizational systems and naturally occurring quality problems. Finally, adaptive quality rules could dynamically adjust validation thresholds according to changing data distributions, business requirements, and historical quality patterns.

8. CONCLUSION

Enterprise organizations increasingly depend on large and heterogeneous data ecosystems, making reliable data-quality management a fundamental requirement for analytics, operational processes, and data-driven decision-making. Conventional approaches frequently address quality assessment, remediation, governance, and lineage as separate activities, limiting their ability to provide continuous and scalable quality management.

This study proposed an automated data quality and governance framework that integrates enterprise data ingestion, metadata discovery, automated profiling, multidimensional quality validation, anomaly detection, governance-policy enforcement, automated remediation, revalidation,

lineage, and operational monitoring within a unified architecture. The design treats data quality as an integral component of the data engineering lifecycle rather than as an isolated preprocessing task.

The framework was evaluated through a controlled experimental methodology using enterprise-style datasets and systematically injected quality defects. The evaluation examined detection performance, data-quality improvement, remediation success, governance compliance, lineage completeness, throughput, latency, and resource utilization across progressively larger workloads and different processing configurations.

The experimental results showed strong detection performance, with an overall precision of 96.2%, recall of 94.8%, and F1-score of 95.5%. Automated remediation achieved a 95.5% success rate, while governance compliance and lineage completeness reached 97.4% and 97.9%, respectively. The scalability experiments also demonstrated increasing processing throughput as worker parallelism increased, reaching 7,420 records/s with eight workers under the evaluated benchmark configuration.

From a data engineering perspective, the framework demonstrates how quality controls can be embedded directly within scalable processing pipelines. From a governance perspective, integrating policy enforcement and lineage provides greater traceability and accountability for quality-related decisions.

Future implementation across real enterprise environments, cloud-native infrastructures, and distributed data ecosystems can further evaluate the framework's generalizability and operational applicability.

REFERENCES

- [1] Scannapieco, M., & Batini, C. (2006). Data quality: concepts, methodologies and techniques. Springer.
- [2] Pipino, L. L., Lee, Y. W., & Wang, R. Y. (2002). Data quality assessment. Communications of the ACM, 45(4), 211-218.
- [3] International Organization for Standardization. (2008). ISO/IEC 25012:2008: Software engineering – Software product Quality Requirements and Evaluation (SQuaRE) – Data quality model. ISO. <https://www.iso.org/standard/35736.html>
- [4] International, DAMA. (2017). DAMA-DMBOK: Data management body of knowledge. Technics Publications, LLC.
- [5] 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.
- [6] Chandran, D. R., & Gupta, V. (2022). A short review of the literature on automatic data quality. Journal of Computer and Communications, 10(5), 55-73.
- [7] Ridzuan, F., & Zainon, W. M. N. W. (2024). A review on data quality dimensions for big data. Procedia computer science, 234, 341-348.
- [8] Valverde, C., Marotta, A., Panach, J. I., & Vallespir, D. (2022). Towards a model and methodology for evaluating data quality in software engineering experiments. Information and Software Technology, 151, 107029.
- [9] 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.
- [10] Daraio, C., Di Leo, S., & Scannapieco, M. (2022). Accounting for quality in data integration systems: a completeness-aware integration approach. Scientometrics, 127(3), 1465-1490.
- [11] Timmerman, Y., Nasfi, R., De Tré, G., Pattyn, F., & Bronselaer, A. (2023). Cost-based analysis of the impact of data completeness and representational consistency. Decision Support Systems, 175, 114044.

- [12] Priestley, M., O'donnell, F., & Simperl, E. (2023). A survey of data quality requirements that matter in ML development pipelines. *ACM Journal of Data and Information Quality*, 15(2), 1-39.
- [13] Patel, H., Guttula, S., Gupta, N., Hans, S., Mittal, R. S., & N, L. (2023). A data-centric AI framework for automating exploratory data analysis and data quality tasks. *ACM journal of data and information quality*, 15(4), 1-26.
- [14] Krasikov, P., & Legner, C. (2023). A method to screen, assess, and prepare open data for use. *ACM Journal of Data and Information Quality*, 15(4), 1-25.
- [15] Mecca, G., Papotti, P., Santoro, D., & Veltri, E. (2024). BUNNI: learning repair actions in rule-driven data cleaning. *ACM Journal of Data and Information Quality*, 16(2), 1-31.
- [16] Peng, J., Shen, D., Tang, N., Liu, T., Kou, Y., Nie, T., ... & Yu, G. (2022). Self-supervised and interpretable data cleaning with sequence generative adversarial networks. *Proceedings of the VLDB Endowment*, 16(3), 433-446.
- [17] Simonini, G., Zecchini, L., Bergamaschi, S., & Naumann, F. (2022). Entity resolution on-demand. *Proceedings of the VLDB Endowment*, 15(7), 1506-1518.
- [18] Grafberger, S., Guha, S., Groth, P., & Schelter, S. (2023). mlwhatif: What if you could stop re-implementing your machine learning pipeline analyses over and over?. *Proceedings of the VLDB Endowment*, 16(12), 4002-4005.
- [19] Qian, Z., Ding, X., Wang, C., & Wang, H. (2025). DemandClean: A Multi-Objective Learning Framework for Balancing Model Tolerance to Data Authenticity and Diversity. *Proceedings of the VLDB Endowment*, 18(12), 5339-5342.
- [20] Mirzaie, M., Behkamal, B., Allahbakhsh, M., Paydar, S., & Bertino, E. (2023). State of the art on quality control for data streams: A systematic literature review. *Computer Science Review*, 48, 100554.
- [21] Behringer, M., Hirmer, P., Villanueva, A., Rapp, J., & Mitschang, B. (2024). Enhancing Data Trustworthiness in Explorative Analysis: An Interactive Approach for Data Quality Monitoring. *SN Computer Science*, 5(5), 465.
- [22] Li, N., Qi, Y., Li, C., & Zhao, Z. (2024). Active learning for data quality control: A survey. *ACM Journal of Data and Information Quality*, 16(2), 1-45.
- [23] Zhu, J., Xu, G., Lorenzi, F., Glavic, B., & Salimi, B. (2025). Stress-Testing ML Pipelines with Adversarial Data Corruption. *arXiv preprint arXiv:2506.01230*.
- [24] Bližňák, K., Munk, M., & Pilková, A. (2024). A systematic review of recent literature on data governance (2017–2023). *IEEE Access*, 12, 149875-149888.
- [25] 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.
- [26] Jahnke, N., & Otto, B. (2023). Data catalogs in the enterprise: applications and integration. *Datenbank-Spektrum*, 23(2), 89-96.
- [27] Eichler, R., Gröger, C., Hoos, E., Stach, C., Schwarz, H., & Mitschang, B. (2023). Introducing the enterprise data marketplace: a platform for democratizing company data. *Journal of Big Data*, 10(1), 173.
- [28] Chen, Y., Zhao, Y., Li, X., Zhang, J., Long, J., & Zhou, F. (2024). An open dataset of data lineage graphs for data governance research. *Visual Informatics*, 8(1), 1-5.
- [29] Yulfitri, A., Sensuse, D. I., Sumirat Hidayat, D., Suryono, R. R., Kautsarina, Purwaningsih, E. H., & Prabuwnono, A. S. (2024). The role of data governance in a high-level approach of data migration to open data. *Business Information Review*, 41(3), 94-103.
- [30] Karkošková, S. (2023). Data governance model to enhance data quality in financial institutions. *Information Systems Management*, 40(1), 90-110.
- [31] Janssen, M., Brous, P., Estevez, E., Barbosa, L. S., & Janowski, T. (2020). Data governance: Organizing data for trustworthy Artificial Intelligence. *Government information quarterly*, 37(3), 101493.
- [32] Yaman, B., Thompson, K., Fahey, F., & Brennan, R. (2024). LinkedDataOps: quality oriented end-to-end geospatial linked data production governance. *Semantic Web*, 15(2), 555-581.
- [33] Syed, R., Eden, R., Makasi, T., Chukwudi, I., Mamudu, A., Kamalpour, M., ... & Myers, T. (2023). Digital health data quality issues: systematic review. *Journal of medical Internet research*, 25, e42615.
- [34] Lighterness, A., Adcock, M., Scanlon, L. A., & Price, G. (2024). Data quality-driven improvement in health care: systematic literature review. *Journal of Medical Internet Research*, 26, e57615.
- [35] Ghalavand, H., Shirshahi, S., Rahimi, A., Zarrinabadi, Z., & Amani, F. (2024). Common data quality elements for health information systems: a systematic review. *BMC Medical Informatics and Decision Making*, 24(1), 243.