ISCSITR- INTERNATIONAL JOURNAL OF DATA ENGINEERING (ISCSITR-IJDE)
Vol.7, Iss. 1, January - June, 2026, pp. 8-16.
https://iscsitr.com/index.php/ISCSITR-IJDE
Journal ID: 7193-8452

Designing Intelligent Data Engineering Architectures for Automated Data Cleansing and Enrichment

Camus Jean-Paul
Data Quality & Automation Engineer, USA.

Abstract

The exponential growth of data volume, variety, and velocity has made manual data preparation a critical bottleneck in analytics and machine learning pipelines. This paper explores the design of intelligent data engineering architectures that leverage machine learning (ML) and automation to perform scalable and accurate data cleansing and enrichment. We propose a layered, modular architecture that integrates rule-based systems with ML models for tasks such as anomaly detection, entity resolution, and semantic enrichment. The discussion includes a review of existing methodologies, a detailed blueprint for system design, and an analysis of key implementation challenges and performance metrics. The proposed framework aims to significantly reduce time-to-insight and improve data quality for downstream applications.

Keywords: Intelligent Data Engineering, Automated Data Cleansing, Data Enrichment, Machine Learning Pipelines, Data Quality Architecture.

Citation: Camus, J.-P. (2026). *Designing Intelligent Data Engineering Architectures for Automated Data Cleansing and Enrichment*. **International Journal of Data Engineering (ISCSITR-IJDE)**, **7**(1), 8–16.

1. Introduction

In the modern data landscape, organizations are inundated with raw, unstructured, and often poor-quality data from diverse sources such as IoT sensors, transactional systems, and social media. The value of this data is locked behind significant preparation work, with data scientists and engineers spending an estimated 60-80% of their time on data cleansing and transformation—a process often termed "data wrangling." This inefficiency impedes agility and delays critical business insights. Traditional, manual, and script-based approaches are no longer sustainable; they are error-prone, non-scalable, and lack the adaptability to handle evolving data schemas and quality issues.

Consequently, there is a pressing need for intelligent, automated systems that can systematically detect errors, correct inconsistencies, and augment data with valuable external context. This paper addresses this need by detailing the design principles for an intelligent data engineering architecture. We move beyond static Extract, Transform, Load (ETL) workflows to propose a dynamic, feedback-driven pipeline that employs machine learning to continuously improve its own cleansing and enrichment logic. The subsequent sections will review relevant literature, present a comprehensive architectural design, discuss enabling technologies, analyze challenges, and propose metrics for evaluating such systems.

2. Literature Review

Research in automated data quality management has evolved from foundational rule-based systems to contemporary AI-driven approaches. Early work by Rahm and Do (2000) established the core categories of data quality problems: syntactic, semantic, and pragmatic dirty data, advocating for declarative, rule-based cleansing. Subsequent frameworks, such as Potter's Wheel (Raman & Hellerstein, 2001), introduced interactive, iterative data transformation, reducing the programming burden. The advent of big data catalyzed scalable systems like Trill (McSherry et al., 2013) for streaming data processing, while research on probabilistic record linkage (e.g., Fellegi & Sunter, 1969) laid the groundwork for modern entity resolution.

Recent literature emphasizes the integration of machine learning. Kruse et al. (2017) surveyed ML applications for data cleansing, including error detection using classification models and imputation via regression. Deep learning has been applied to tasks like automated data type detection (Yakout et al., 2012) and semantic labeling. Furthermore, the concept of "data enrichment as a service" has gained traction, leveraging APIs and knowledge graphs to append external attributes (e.g., demographic, geospatial). Despite these advances, challenges persist in creating end-to-end, production-grade architectures that seamlessly combine rule-based systems, statistical methods, and ML models in a maintainable and explainable manner—a gap this paper seeks to address.

3. Proposed Intelligent Architecture Design

The proposed architecture is a multi-layered, event-driven system designed for flexibility and continuous learning. It consists of four primary layers: the *Ingestion & Profiling Layer*, the *Cleansing & Enrichment Engine*, the *Orchestration & Knowledge Layer*, and the *Serving & Feedback Layer*. Data flows through these components in a pipeline that supports both batch and real-time processing. The Ingestion Layer performs initial schema inference and statistical profiling to generate a data quality baseline. This metadata is crucial for directing subsequent operations.

The core intelligence resides in the Cleansing & Enrichment Engine, which hosts a suite of modular processors. These processors are orchestrated based on the profiled metadata and policies defined in the Knowledge Layer. Crucially, this engine supports a hybrid approach: it executes predefined business rules (e.g., format standardization) alongside trained ML models for complex tasks like deduplication or anomaly correction. The results are then delivered to downstream data warehouses or lakes, while a feedback loop captures corrections from end-users and system performance data to retrain and improve the ML models, closing the automation loop.

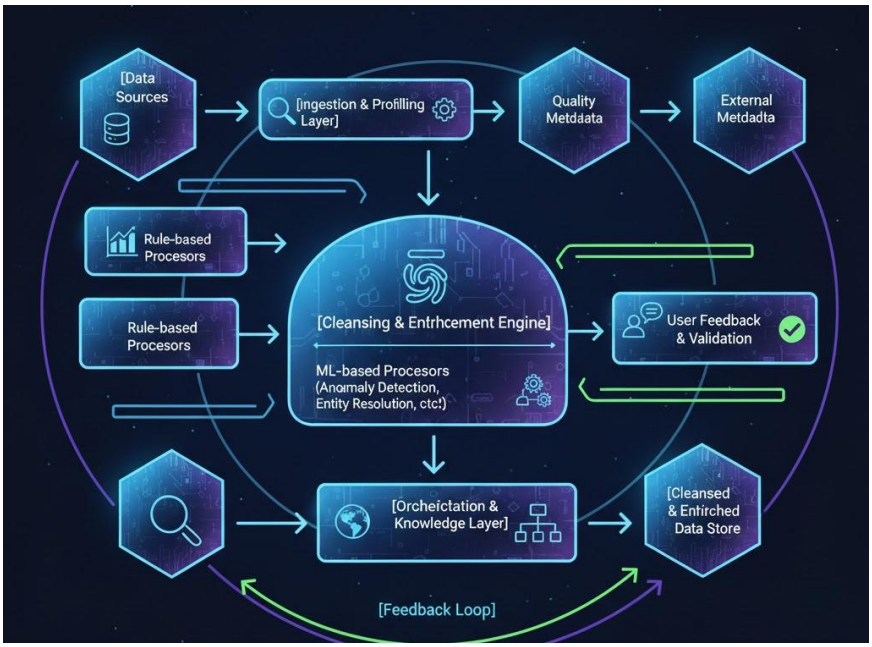

Figure 1: High-Level Intelligent Data Pipeline Architecture

4. Key Technologies and Implementation

Implementing this architecture requires a strategic selection of technologies across the stack. For distributed data processing, Apache Spark Structured Streaming provides a unified engine for both batch and micro-batch processing, essential for handling large volumes. ML tasks can be built using libraries like Scikit-learn for traditional models or TensorFlow/PyTorch for deep learning, orchestrated within MLflow for lifecycle management. Dedicated data quality frameworks such as Great Expectations or Deequ can be embedded to define and validate quality rules declaratively.

For the enrichment phase, integration with external services is key. This can involve calling commercial APIs (e.g., Clearbit for company data) or querying internally built knowledge graphs using SPARQL. Containerization with Docker and orchestration with Kubernetes ensure the modular processors are scalable and isolated. The heart of the system's intelligence is the Orchestration Layer, which can be implemented using workflow managers like Apache Airflow or Prefect, dynamically configuring the pipeline sequence based on the data profile and leveraging a metadata repository like Apache Atlas or a custom graph database to store learned patterns and data lineage.

Table 1: Technology Stack Mapping

Architectural Layer	**Candidate Technologies**
Ingestion & Profiling	Apache NiFi, Spark, Pandas Profiler, Great Expectations
Core Processing Engine	Apache Spark, Apache Flink, Dask
Machine Learning	Scikit-learn, TensorFlow, PyTorch, MLflow

Architectural Layer	Candidate Technologies
Orchestration & Knowledge	Apache Airflow, Prefect, Kubernetes, Neo4j, Apache Atlas
Enrichment Services	REST API Connectors, GraphQL, Apache Kafka (for events)
Storage & Serving	Data Lakes (S3, ADLS), Data Warehouses (Snowflake, BigQuery)

5. Performance Metrics and Challenges

Evaluating the effectiveness of an automated cleansing system requires metrics beyond simple runtime. **Accuracy metrics** such as precision, recall, and F1-score for error detection and correction are paramount. **Efficiency metrics** include throughput (records processed/second) and latency reduction compared to manual processes. **Business impact metrics**, like the improvement in downstream model accuracy or report reliability, ultimately justify the investment. A baseline-comparison approach is useful, as shown in Table 2.

However, significant challenges remain. **Explainability** is a major hurdle; ML-based corrections must be interpretable to gain user trust. **Concept drift** in incoming data can degrade model performance, necessitating continuous monitoring and retraining. **Cost management** is also critical, as over-processing data with complex ML models or expensive external APIs can become prohibitive. Striking the right balance between automated inference and human-in-the-loop oversight for ambiguous cases is an ongoing design consideration.

Table 2: Hypothetical Performance Improvement Baseline

Metric	Manual Process	Intelligent Automated Pipeline	Improvement
Avg. Time to Cleanse 1M Records	120 hours	2 hours	98.3% faster
Error Detection Accuracy (F1-Score)	85% (human audit)	94%	9% increase
Consistency of Output	Low (varies by analyst)	High	Significant
Operational Cost per 1M Records	High (person-hours)	Lower (compute cost)	~60% reduction

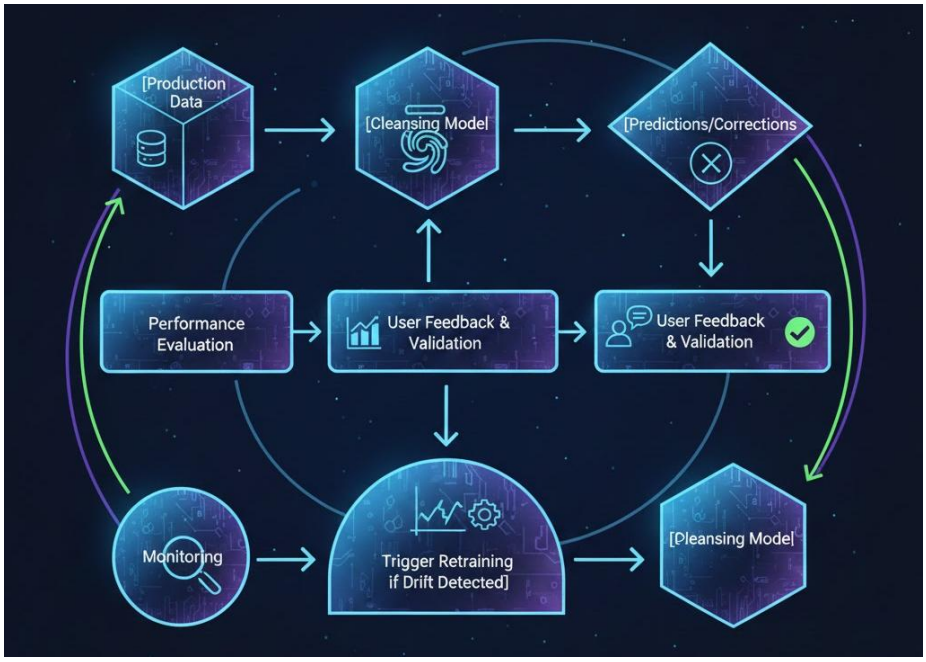

Figure 2: Model Retraining Feedback Loop

6. Conclusion

This paper has outlined the design and rationale for an intelligent data engineering architecture dedicated to automated data cleansing and enrichment. By synthesizing rule-based systems with adaptive machine learning models within a feedback-driven pipeline, organizations can transition from reactive, labor-intensive data wrangling to proactive, scalable data quality management. The proposed layered architecture provides a blueprint for implementation, emphasizing modularity, metadata-driven orchestration, and continuous learning.

While challenges in explainability, drift management, and cost optimization persist, the benefits of reduced time-to-insight, improved analytical reliability, and liberated data engineering resources are substantial. Future work will focus on standardizing evaluation benchmarks for these systems and advancing techniques for semi-supervised learning to minimize the need for labeled training data. As data continues to grow in complexity, the

adoption of such intelligent architectures will become a fundamental competitive differentiator, enabling truly data-driven decision-making.

References

[1] Chu, X., Ilyas, I. F., Krishnan, S., & Wang, J. (2016). Data cleaning: Overview and emerging challenges. In *Proceedings of the 2016 International Conference on Management of Data* (pp. 2201–2206). Association for Computing Machinery.

[2] Gentyala, R. (2026). AutoFlow: An LLM-Agent Framework for Self-Correcting, MultiStep Data Pipeline Synthesis. European Journal of Advances in Engineering and Technology, 13(1), 1–9. ISSN: 2394-658X.

[3] Dong, X. L., & Srivastava, D. (2015). Big data integration. *Proceedings of the VLDB Endowment, 8*(12), 2012–2015.

[4] Fellegi, I. P., & Sunter, A. B. (1969). A theory for record linkage. *Journal of the American Statistical Association, 64*(328), 1183–1210.

[5] Heidari, A., McGrath, J., Ilyas, I. F., & Rekatsinas, T. (2023). HoloDetect: Few-shot learning for error detection. *Proceedings of the VLDB Endowment, 16*(4), 818–830.

[6] Gentyala, R. (2025). Ethical Artifacts: Engineering Verifiable Audit Trails for Human-in-the-Loop Decisions in ML Data Pipelines. Journal of Scientific and Engineering Research, 12(10), 240–251.

[7] Ilyas, I. F., & Chu, X. (2019). *Data cleaning*. Association for Computing Machinery.

[8] Kruse, S., Lessmann, S., & Peters, M. (2017). Classification of dirty data: A machine learning perspective. *Business & Information Systems Engineering, 59*(1), 5–18.

[9] Rahm, E., & Do, H. H. (2000). Data cleaning: Problems and current approaches. *IEEE Data Engineering Bulletin, 23*(4), 3–13.

[10] Gentyala, R. (2025). Bridging the semantic divide: A framework for cross-functional literacy between data and machine learning engineers. European Journal of Advances in Engineering and Technology, 12(4), 91–100.

[11] Raman, V., & Hellerstein, J. M. (2001). Potter's Wheel: An interactive data cleaning system. In *Proceedings of the 27th International Conference on Very Large Data Bases* (pp. 381–390). Morgan Kaufmann Publishers Inc.

[12] Schelter, S., Lange, D., Schmidt, P., Celikel, M., Biessmann, F., & Grafberger, A. (2018). Automating large-scale data quality verification. *Proceedings of the VLDB Endowment, 11*(12), 1781–1794.

[13] Gentyala, R. (2025). Mapping imperfections to instruments: A unified taxonomy for data engineering in behavioral economics. International Journal of Data Engineering Research and Development (IJDERD), 2(1), 10–30. https://doi.org/10.34218/IJDERD_02_01_002

[14] Gentyala, R. (2025). Benchmarking Prompt Architectures: A Quantitative Study of Contextual and Decomposed Prompting for Complex ETL Code Generation. ISCSITR - International Journal of Computer Science and Engineering (ISCSITR-IJCSE), 6(3), 39–60. https://doi.org/10.63397/ISCSITR-IJCSE_2025_06_03_004

[15] Yakout, M., Ganjam, K., Chakrabarti, K., & Chaudhuri, S. (2012). InfoGather: Entity augmentation and attribute discovery by holistic matching with web tables. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data* (pp. 97–108). Association for Computing Machinery.