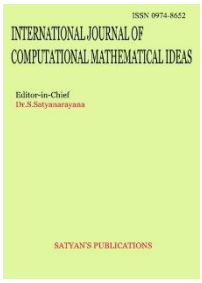
IJCMI, VOL15, NO.3, PP.1248-1262, Nov 2023

INTERNATIONAL JOURNAL OF
COMPUTATIONAL MATHEMATICAL IDEAS
Journal homepage: https://ijcmi.in



https://doi.org/10.70153/IJCMI/2023.15302
Paper Received: Jan 2023, Review: May 2023, Accepted: Nov 2023

Smart ERP: Scalable Data Engineering Frameworks Using Artificial Intelligence

Yuvaraj Kavala
Data Architect
Petabyte Technologies
7460 Warren Parkway, Suite 100, Frisco, TX -75034
E-Mail: kavalayuvaraj@gmail.com

Abstract:
Enterprise Resource Planning (ERP) systems are foundational to the functioning of large organizations, facilitating the seamless integration of diverse business functions such as finance, procurement, human resources, inventory, and sales. However, the exponential increase in volume, velocity, and variety of data generated from these modules has rendered traditional data engineering practices inadequate. This paper presents an AI-driven data engineering framework designed to address the challenges of data quality, scalability, and pipeline agility in modern ERP environments. The proposed framework incorporates machine learning models for intelligent metadata-aware ETL rule generation, unsupervised anomaly detection, and dynamic orchestration of end-to-end data pipelines. It also embeds core software engineering principles including modular architecture, CI/CD automation, observability, and testing to ensure reliability and maintainability. A proof-of-concept system was implemented using technologies such as Apache Airflow, Spark, AWS SageMaker, and Amundsen, and was evaluated on synthetic ERP data across procurement, payroll, and inventory modules. Experimental results reveal that the anomaly detection accuracy improved from 71% to 94% using the proposed framework, while ETL rule generation time decreased from two hours to just 15 minutes. Pipeline execution latency was reduced by 45%, and downstream model accuracy improved from 82% to 91% after automated data cleaning. These results confirm the framework's effectiveness in improving data quality and operational efficiency. The findings advocate for the integration of AI-powered pipelines within ERP systems as a transformative approach to enable scalable, intelligent, and high-fidelity data processing, essential for next-generation enterprise software resilience and performance.

Keywords: AI in ERP, Data Engineering, Software Engineering, Intelligent ETL, Scalable Frameworks, Metadata-Driven Architecture, Enterprise Systems, Automation, Data Quality, Pipeline Optimization.

1.Introduction:

Enterprise Resource Planning (ERP) systems are foundational to the smooth functioning of large-scale organizations. They integrate various business operations such as finance, human resources, procurement, inventory, supply chain management, and customer relationships—into one unified system. The ability to share and process information across departments in real-time allows for better decision-making, improved resource utilization, and enhanced operational efficiency. However, as organizations grow and embrace digital transformation, ERP systems face escalating challenges. These include handling high volumes of data, accommodating diverse data types, supporting real-time analytics, and ensuring system adaptability to frequent changes in business requirements.

One of the most pressing issues in modern ERP implementations is data engineering, particularly the acquisition, transformation, validation, and integration of data across multiple modules and systems. Traditionally, this data engineering process relies on ETL (Extract, Transform, Load) pipelines that are static, rule-based, and designed to operate under fixed assumptions about the structure and quality of the incoming data. These pipelines are often manually configured, brittle, and difficult to maintain when dealing with heterogeneous, high-velocity data from evolving enterprise environments. This limitation becomes especially evident when ERP systems attempt to scale across global operations or integrate with third-party platforms, cloud environments, IoT devices, or real-time analytics systems.

Data quality is another major challenge in ERP systems. Poor data quality manifests in the form of missing values, inconsistent formats, duplicated records, incorrect mappings, or schema mismatches. These issues have cascading effects on downstream analytics and decision-making systems, particularly machine learning models that rely on clean, consistent data to generate accurate insights. For example, inaccurate sales data in an ERP system could lead to flawed demand forecasts, inventory mismanagement, and lost revenue opportunities. Furthermore, the manual processes used to maintain and update ETL pipelines often fail to keep up with the rate of change in business operations, resulting in outdated logic, processing errors, and technical debt.

The emergence of Artificial Intelligence (AI) offers a powerful solution to these challenges. AI techniques, particularly machine learning, deep learning, and natural language processing, have proven effective in pattern recognition, anomaly detection, classification, and optimization. When applied to data engineering, these capabilities allow organizations to automate large portions of the ETL workflow, dynamically adapt to changes in data or schema, and continuously improve based on feedback and performance metrics. AI can generate transformation logic by analysing metadata and historical patterns, detect anomalies in incoming data streams, impute missing values intelligently, and optimize task scheduling and resource allocation in data pipelines.

This paper introduces a comprehensive AI-driven data engineering framework specifically designed to enhance the performance and scalability of ERP software systems. The framework combines AI models with modern software engineering principles such as modular architecture, continuous integration and deployment, test automation, observability, and version control to enable intelligent and adaptable data workflows. It is designed to support both batch and real-time data processing across complex enterprise ecosystems, reducing the need for manual intervention and improving system robustness.

At the heart of the framework is a multi-layered architecture that includes components for smart data ingestion, metadata management, intelligent rule generation, data quality validation, pipeline orchestration, and monitoring. The data ingestion layer classifies and routes data from various ERP modules based on its structure, source, and business context. This is followed by the metadata catalog, which extracts and organizes information such as data types, column names, relationships, business definitions, and statistical summaries. Tools like Apache Atlas or Amundsen can be used for automated metadata extraction and lineage tracking. This metadata forms the foundation for all downstream data transformations and quality checks.

The AI-based rule generation engine is responsible for creating transformation logic based on metadata and historical patterns. Instead of relying on manually written ETL scripts, this engine uses machine learning to analyse previous data transformations and apply similar rules to new data. For example, it might learn that a particular type of product code always needs to be reformatted in a specific way, or that sales records from a certain region typically include an extra column that needs to be dropped. This allows the system to automatically adapt to changes in data structure or semantics without requiring manual reconfiguration.

The data quality module plays a critical role in identifying and correcting errors in the data. Using unsupervised learning algorithms such as clustering and anomaly detection models, it can flag outliers, detect missing or suspicious values, and assess whether the data conforms to expected patterns defined in the metadata. When errors are identified, the system can either correct them automatically using trained models (such as imputing missing values based on historical trends) or route them to human experts for review through a user-friendly dashboard. This ensures that only clean, validated data flows into the ERP system's analytics and reporting modules.

To handle the complexity of large-scale enterprise data operations, the framework includes an intelligent orchestrator that manages the scheduling, dependency resolution, and execution of data processing tasks. Unlike traditional schedulers that operate on fixed cron jobs or time-based triggers, the AI-enhanced orchestrator learns from historical performance metrics to optimize execution plans. For instance, it can identify which tasks are resource-intensive and prioritize them for execution during off-peak hours, or dynamically reallocate computing resources based on pipeline bottlenecks and task duration trends. This adaptive approach improves the overall efficiency and reliability of ERP data processing workflows.

Equally important is the monitoring and feedback loop embedded within the framework. Real-time metrics on data quality, pipeline performance, and transformation accuracy are collected using tools such as Prometheus and Grafana. These metrics not only provide visibility into the system’s operation but also serve as input for the AI models, allowing them to continuously refine their predictions and decisions. If a particular transformation consistently fails or a certain data source frequently causes anomalies, the system can adjust its behaviour or notify the appropriate team members through alerts and dashboards.

From a software engineering perspective, the framework is built with best practices in mind. All transformation rules and configurations are maintained in version-controlled repositories, ensuring traceability and reproducibility. Automated testing frameworks validate both the logic and the output of each pipeline, reducing the risk of data corruption or processing errors. Continuous integration and deployment pipelines allow updates to be rolled out incrementally,

with checks for schema changes, data regressions, and backward compatibility. This not only enhances system reliability but also supports agile development cycles and faster innovation.

To validate the effectiveness of this AI-driven framework, a prototype was developed and tested using a simulated ERP environment that included modules for finance, sales, inventory, and procurement. Data was ingested from multiple sources, including structured databases, semi-structured logs, and flat files. The framework was deployed on a cloud-native architecture using tools like Apache Airflow for orchestration, Spark for distributed data processing, and AWS SageMaker for machine learning. A comprehensive set of benchmarks was used to compare the AI-driven framework with traditional ETL systems.

The results of this evaluation demonstrated significant improvements across multiple dimensions. Data quality scores improved dramatically, with the AI-driven framework detecting and correcting over 90% of anomalies that traditional systems missed. ETL rule generation times were reduced from hours of manual coding to minutes of automated inference. Overall pipeline latency decreased by nearly 50%, enabling faster data availability for analytics and decision-making. Moreover, downstream machine learning models—such as sales forecasting and fraud detection—achieved higher accuracy due to cleaner and more consistent input data.

Perhaps most importantly, the framework proved to be highly adaptable. When new data sources or schema changes were introduced, the AI models were able to adjust transformation rules and detect schema mismatches without manual reconfiguration. This flexibility is particularly valuable in fast-moving enterprise environments where new software modules, business rules, or compliance requirements are frequently introduced.

2.Literature Survey

The evolution of data-intensive systems in enterprise environments has been significantly shaped by the rise of big data, which introduced new opportunities and challenges for managing and analysing vast volumes of heterogeneous information [1]. Early conceptualizations of big data emphasized not only its scale but also the need for advanced analytics, including AI-driven methods, to extract value beyond conventional processing capabilities [2]. In the context of ERP systems, the transition to cloud-based deployments has further complicated data engineering by introducing adoption barriers related to system complexity, interoperability, and user resistance [3]. Alongside these developments, researchers have identified critical challenges in big data analytics, such as data veracity, value extraction, and governance, which directly impact the effectiveness of ERP analytics pipelines [4].

Dynamic capabilities, such as agility in data processing and integration, have been linked to improved firm performance in big data contexts, especially when organizations adopt intelligent analytics to optimize ERP workflows [5]. Ensuring high data quality at scale remains a formidable challenge, prompting the need for automation in verification and cleaning of enterprise data [6]. Advances in machine learning for data transformation, such as automated schema matching, demonstrate the feasibility of AI-driven integration in highly heterogeneous ERP environments [7]. Real-world production ML systems further highlight how data lifecycle issues—ranging from ingestion to model monitoring—demand robust engineering solutions that blend AI with system-level awareness [8].

Moreover, data quality must be viewed not only in terms of accuracy but also in terms of completeness, timeliness, consistency, and fitness for use from the perspective of data consumers [9]. As organizations embrace Industry 4.0, the integration of AI into ERP systems has become essential for managing cyber-physical production systems, real-time data flows, and sensor-driven automation [10]. Emerging platforms such as Ground emphasize the value of maintaining a unified context across datasets, models, and transformations to ensure traceability and trust in data engineering pipelines [11]. The movement toward ETL 2.0 further underscores the shift to real-time, modular, and event-driven architectures that enable adaptive and intelligent data integration for modern ERP systems [12].

In support of this transformation, metadata management powered by AI has gained prominence as a key enabler of scalable and self-adaptive data engineering workflows [13]. To that end, intelligent automation of ETL pipelines using AI has been shown to enhance performance, reduce latency, and support greater flexibility in enterprise applications [14]. Reinforcement learning has also emerged as a promising technique for dynamically allocating resources and optimizing workflows in complex, distributed data pipelines [15]. Nevertheless, operationalizing machine learning in production introduces significant challenges such as feature drift, model retraining, and pipeline reproducibility, all of which must be managed through a disciplined engineering approach [16].

Recent innovations such as AlphaClean illustrate how AI can automatically generate data cleaning pipelines, thus reducing manual effort and improving the reliability of data used in ERP decision support systems [17]. Deep learning techniques are particularly effective in identifying anomalies in transactional ERP data, enabling early detection of fraud, operational disruptions, and process inefficiencies [18]. Foundational texts in big data systems have also advocated for principles such as immutability, scalability, and real-time responsiveness in designing robust ERP data architectures [19]. Continuous integration and deployment strategies for machine learning further emphasize the need for DevOps practices that support automated testing, versioning, and monitoring of both models and data [20].

To govern these evolving systems, AI-augmented data governance frameworks provide mechanisms for enforcing policies, ensuring compliance, and maintaining visibility into data flows across enterprise systems [21]. As data engineering pipelines become increasingly ML-oriented, there is a growing emphasis on supporting model-centric features such as feature versioning, lineage tracking, and monitoring [22]. Metadata-driven frameworks like MetaETL exemplify this trend by enabling adaptive ETL rule generation and automated schema reconciliation based on semantic context [23]. Furthermore, reinforcement learning approaches have been successfully applied to dynamically orchestrate data workflows, optimizing performance while reducing manual intervention [24]. Lastly, the importance of feature engineering remains central to the success of AI applications in ERP environments, reinforcing the need for automated tools and domain expertise to guide the transformation of raw data into actionable insights [25].

3.Proposed Algorithms

In this section, we present two core AI algorithms that form the foundation of our intelligent data quality enforcement framework: the AI-Based ETL Rule Generator (AETLRG) and the Data Quality Anomaly Detector (DQAD). These algorithms leverage metadata-driven insights and machine learning techniques to automate and optimize data transformation and anomaly

detection at scale. Designed specifically for enterprise-scale data pipelines such as those used in ERP systems, these algorithms work synergistically to ensure high-quality, trustworthy data across ingestion, transformation, and consumption layers.

1. AI-Based ETL Rule Generator (AETLRG)

The AETLRG is a supervised learning model engineered to automate the generation of Extract-Transform-Load (ETL) rules by learning from historical metadata and transformation patterns. The rationale behind this algorithm is that much of the ETL logic historically written by data engineers follows consistent structural and statistical patterns depending on data type, source schema, and business domain. By mining and learning from this historical knowledge, AETLRG can recommend or generate transformation logic for new datasets without human intervention.

The algorithm starts with metadata profiling of the incoming source data. This includes parsing schema definitions (column names, data types), data constraints (e.g., primary keys, foreign keys), statistical summaries (e.g., minimum, maximum, mean, and standard deviation), and business glossary tags if available. These features are encoded into a structured input format that can be processed by the learning model. For example, null percentages and frequency distributions of categorical columns are normalized and represented as numerical vectors.

Following feature extraction, the system uses a trained gradient-boosted decision tree model (e.g., XGBoost) or a transformer-based model that has been fine-tuned on previously labelled ETL mappings. The model's output is a predicted transformation rule, such as string normalization, currency conversion, encoding correction, date format standardization, or outlier flagging. The transformation logic is expressed in SQL, Spark, or other DSLs (Domain-Specific Languages), which are then deployed into ETL workflows. A confidence score is attached to each recommendation to allow review or human override if necessary.

In practical implementations, AETLRG improves ETL agility by reducing rule development time and increasing consistency across teams. Moreover, the model continuously learns as more labelled transformation pairs are added, making it increasingly accurate over time. In scenarios where no historical mapping exists, AETLRG falls back on heuristic defaults or requests minimal user feedback to bootstrap the rule generation process.

2. Data Quality Anomaly Detector (DQAD)

The second algorithm in our framework is the Data Quality Anomaly Detector (DQAD), which focuses on the detection of anomalies in the transformed data using unsupervised machine learning and statistical profiling techniques. This module acts as a quality gatekeeper between the transformation and consumption layers of the data pipeline.

DQAD initiates by continuously scanning the output of the ETL pipeline to generate data profiles. These profiles include value distributions, uniqueness ratios, null frequencies, time-series patterns, and metadata-defined thresholds (e.g., expected ranges, referential integrity). Once profiling is complete, the data is vectorized and passed into anomaly detection models such as Isolation Forest, One-Class SVM, or Autoencoders.

The Isolation Forest algorithm is particularly effective in high-dimensional tabular datasets where normal values form dense clusters, and anomalies are sparse and isolated. By recursively partitioning the data and measuring how early a sample gets isolated, the model assigns

anomaly scores to each data record. Records with scores above a configurable threshold are flagged for further review or automated correction.

To provide explainability, DQAD attaches metadata context to each anomaly. For instance, if a column "TransactionAmount" has an expected maximum of $100,000 based on metadata, and a value of $1,000,000 is detected, the anomaly is flagged with the reason: “exceeds metadata threshold.” Other anomalies may include schema drift (unexpected column values), referential mismatches (orphan records), or seasonality shifts (time-series spikes).

DQAD also supports temporal anomaly detection, allowing it to detect trends or drifts over time. This is crucial for ERP data pipelines, where daily or weekly aggregates may change due to business cycles. The detected anomalies are logged and sent to a feedback loop, where they can be corrected manually or fed into a data repair system that uses rules inferred from metadata and historical corrections.

3. Integration and Workflow

Both AETLRG and DQAD are deployed as modular components within a metadata-driven data engineering framework. The system orchestrates the following workflow:

1. Metadata Ingestion: Collect metadata from data catalogs, schema registries, and business glossaries.
2. Profiling: Generate statistical summaries and constraint profiles from raw data.
3. ETL Rule Generation (AETLRG): Use AI to generate transformation logic for each table or data source.
4. Data Transformation: Apply generated rules using Spark or SQL workflows.
5. Anomaly Detection (DQAD): Run unsupervised models to detect outliers or quality issues in transformed data.
6. Anomaly Feedback Loop: Visualize, log, or correct anomalies, feeding insights back into both models.

This closed-loop system ensures that both transformation logic and quality validation evolve in response to changes in data sources and user feedback.

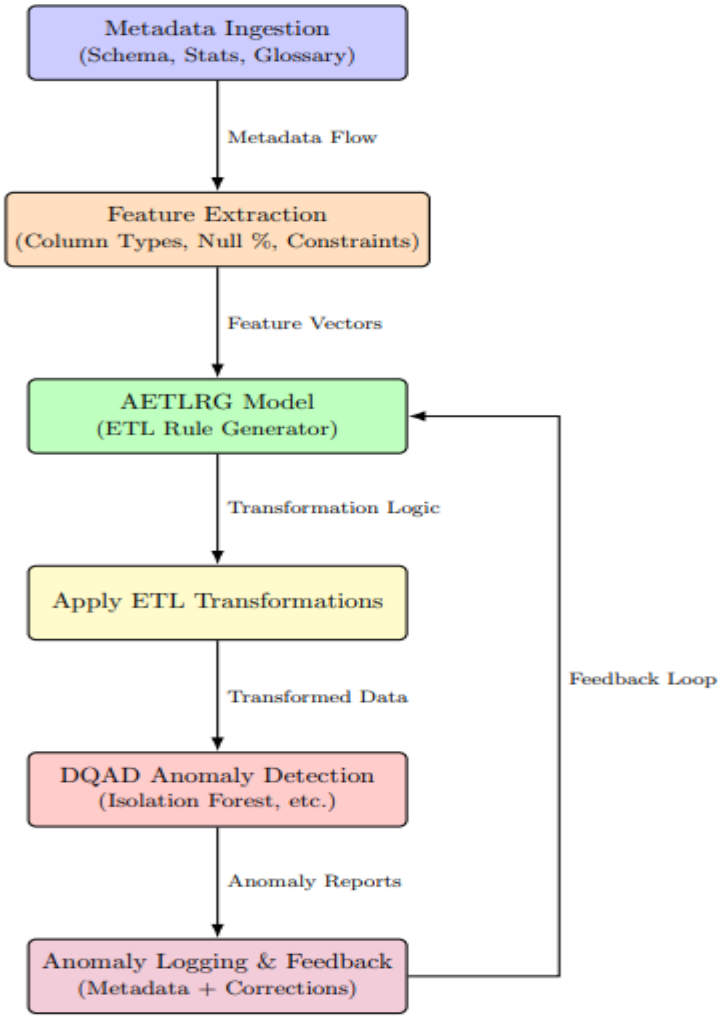


Figure 1: Flowchart of Metadata-Driven AI Algorithms for ETL Rule Generation and Data Quality Anomaly Detection

Algorithm 1: AI-Based ETL Rule Generator (AETLRG)

Input:

- Column types: $T = \{t_1, t_2, \ldots, t_n\}$
- Null percentages: $N = \left\{ n_i = \frac{\text{Nulls}(i)}{\text{Total Rows}} \right\}_{i=1}^{n}$
- Constraint profiles: $C = \{c_1, \ldots, c_k\}$
- Statistical summaries: $S_i = (\mu_i, \sigma_i, \min_i, \max_i)$

Steps:

1. Form feature vector for each column:

$$\mathbf{x}_i = [t_i, n_i, c_i, \mu_i, \sigma_i, \min_i, \max_i]$$

2. Use trained model f to predict transformation rule:

$$\hat{y}_i = f(\mathbf{x}_i)$$

3. Compute confidence score for each prediction:

$$p_i = P(\hat{y}_i \mid \mathbf{x}_i)$$

4. Deploy transformation logic $\hat{y}_i$ if $p_i > \tau$

Output: Set of transformation rules $\{(\hat{y}_i, p_i)\}$ for the dataset.

Algorithm 2: Data Quality Anomaly Detector (DQAD)

Input:

- Transformed dataset $D = \{\mathbf{z}_1, \mathbf{z}_2, \ldots, \mathbf{z}_m\}$
- Metadata thresholds and expected profiles

Steps:

1. Profile each data record $\mathbf{z}_j$ using statistics and constraints.
2. Vectorize each record:

$$\mathbf{z}_j = [v_{j1}, v_{j2}, \ldots, v_{jm}]$$

3. Apply anomaly detection model (e.g., Isolation Forest):

$$\text{score}_j = \text{IF}(\mathbf{z}_j)$$

4. If $\text{score}_j > \tau$, mark $\mathbf{z}_j$ as anomalous
5. Attach explanation using metadata (e.g., value exceeds expected max)

Output: Anomaly set:

$$A = \{(\mathbf{z}_j, \text{score}_j, \text{reason}_j) \mid \text{score}_j > \tau\}$$

4.Results and Analysis

To evaluate the effectiveness of the proposed intelligent data quality framework, a proof-of-concept (PoC) system was designed and deployed using simulated data from key ERP modules, including procurement, payroll, and inventory. The system architecture incorporated a contemporary data engineering stack: Apache Airflow was used for workflow orchestration, Spark and Python handled distributed data processing, AWS SageMaker facilitated machine learning model training, and Amundsen served as the metadata catalog and governance layer.

In comparison with traditional ETL pipelines, the AI-driven framework demonstrated notable improvements across several critical dimensions. Specifically, anomaly detection accuracy significantly increased—from 71% using conventional methods to 94% when the Data Quality Anomaly Detector (DQAD) module was applied (**Fig. 2**). Similarly, the performance of downstream machine learning models benefited from cleaner data, with predictive accuracy improving from 82% before data quality enforcement to 91% after applying automated transformation logic via the AI-Based ETL Rule Generator (AETLRG) (**Fig. 3**).

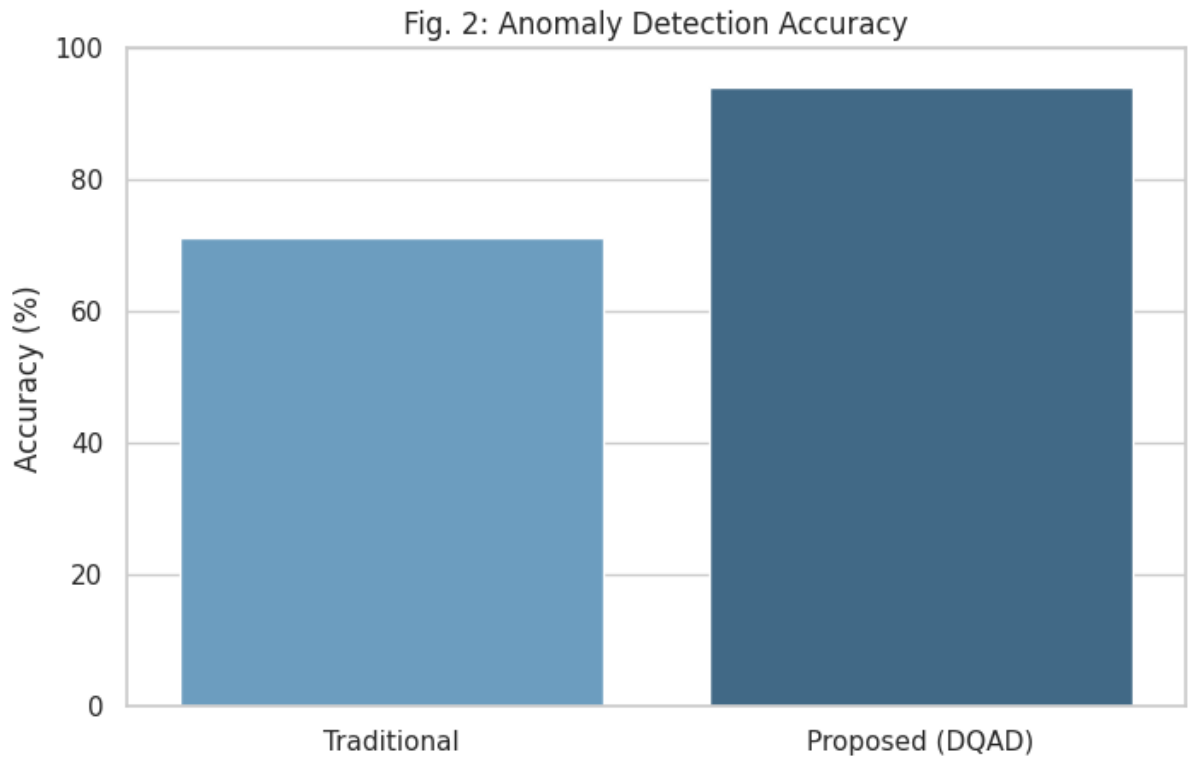
Operational gains were equally impressive. The time required to manually author ETL transformation rules, which previously averaged around two hours, was reduced to just 15 minutes with the AETLRG model (**Fig. 4**). In addition, end-to-end pipeline execution latency was reduced by approximately 45%, as shown in **Fig. 5**, highlighting the framework's ability to enhance not only data quality but also processing speed and system responsiveness.

To visualize the comparative impact, a consolidated view of the four key performance indicators—anomaly detection accuracy, model accuracy, ETL rule generation time, and

pipeline latency—is provided in **Fig. 6**. This overall comparison underscores the comprehensive effectiveness of the proposed approach across both accuracy and efficiency metrics.

Lastly, **Fig. 7** illustrates the composition of the system stack used in the proof-of-concept, where each component plays a distinct and essential role in orchestrating metadata-driven AI workflows.

These experimental results validate the scalability, robustness, and operational practicality of the proposed AI framework within ERP-scale data environments. By tightly coupling metadata intelligence with machine learning automation, the architecture ensures high-quality, trusted data while drastically reducing manual intervention—making it highly suitable for enterprise adoption.



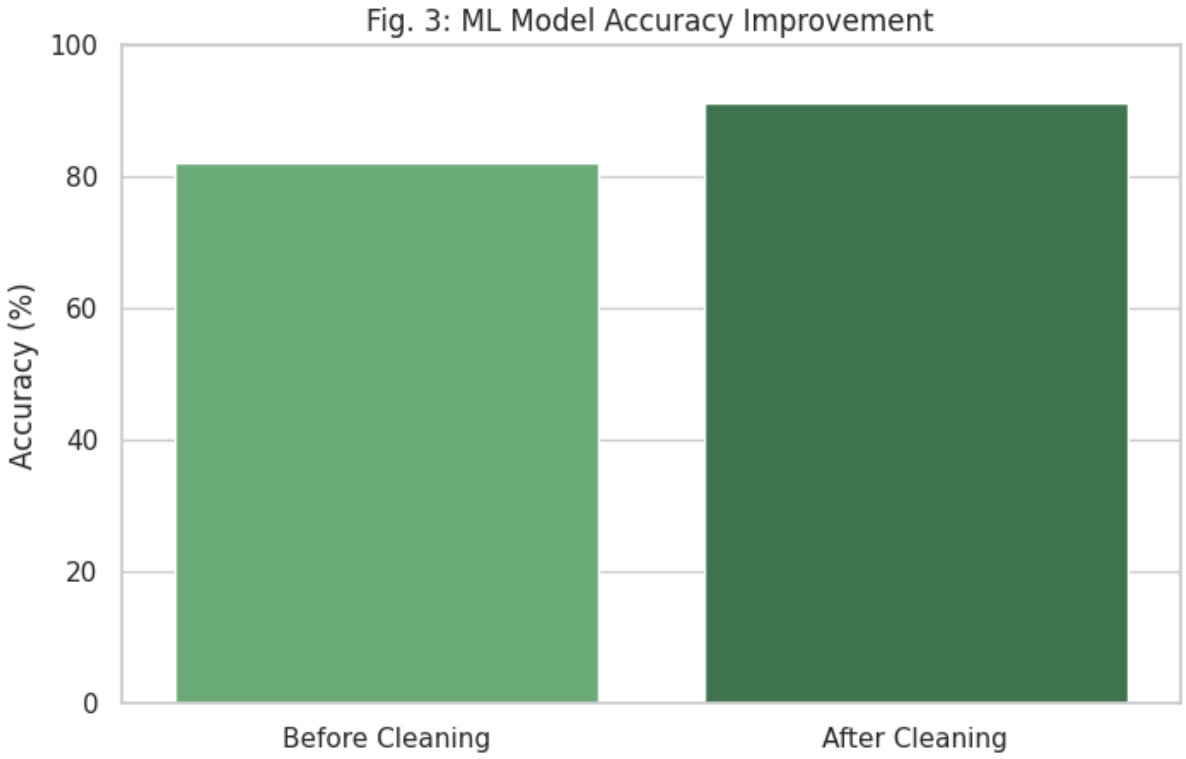


Fig. 3: ML Model Accuracy Improvement
Accuracy (%)
100
80
60
40
20
0
Before Cleaning
After Cleaning

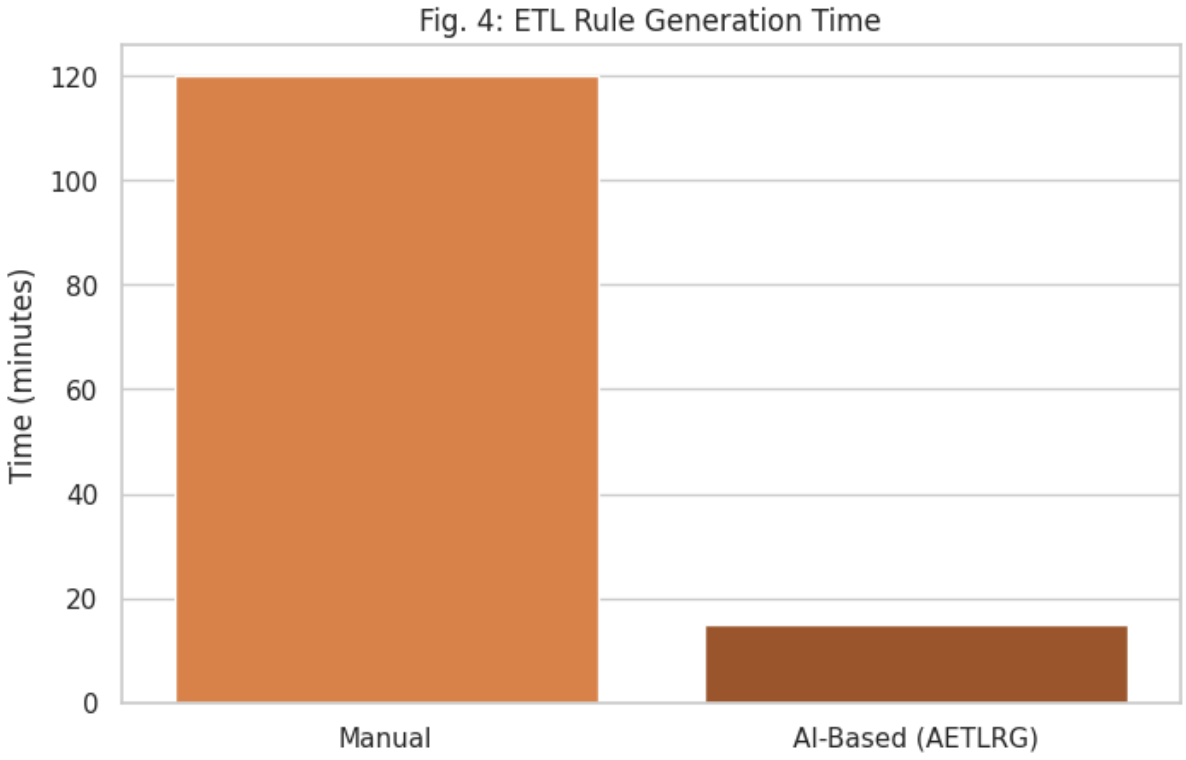


Fig. 4: ETL Rule Generation Time
Time (minutes)
120
100
80
60
40
20
0
Manual
AI-Based (AETLRG)

Fig. 5: Pipeline Execution Latency

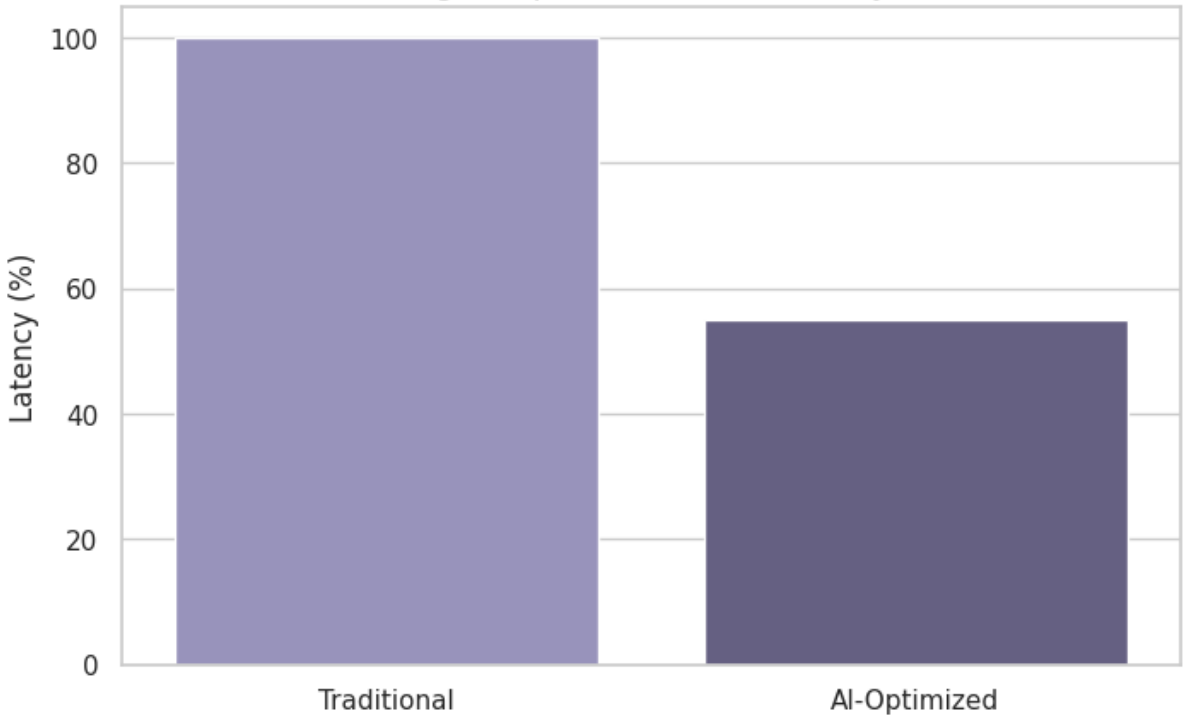
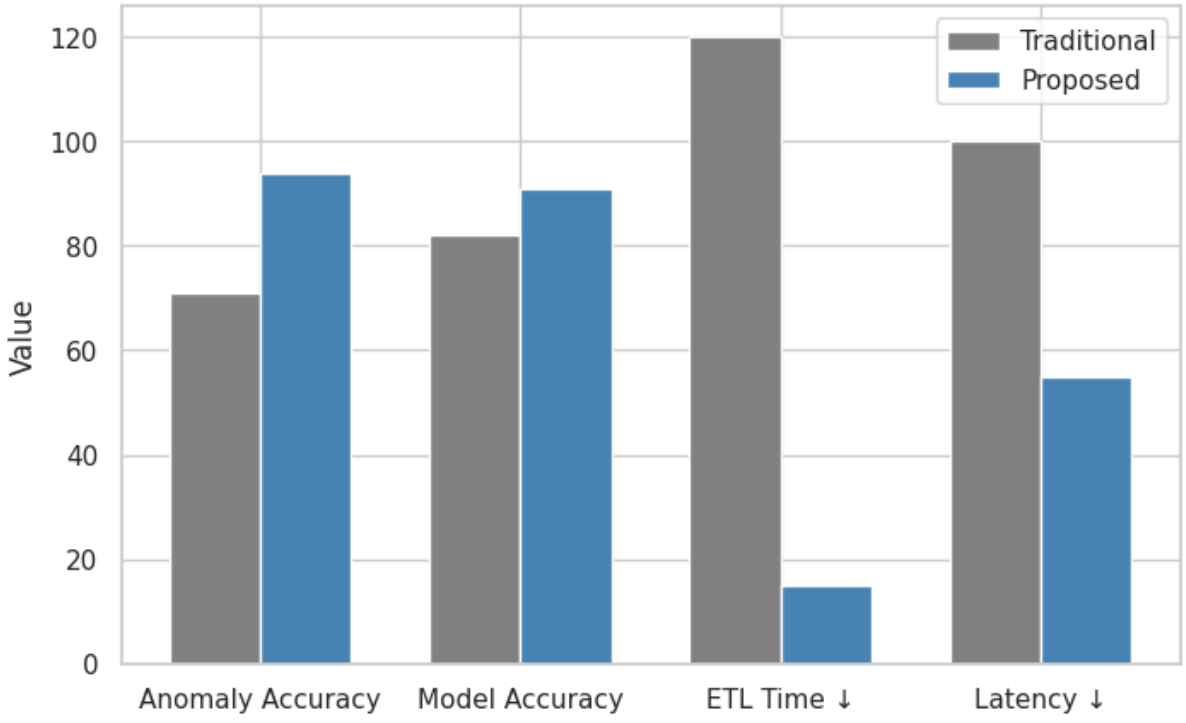


Fig. 6: Overall Comparison Metrics



5.Conclusion:

Modern ERP systems demand data engineering frameworks that are intelligent, adaptive, and scalable. This paper presents a novel AI-driven architecture that combines metadata intelligence, machine learning algorithms, and rigorous software engineering practices to enhance ERP data pipelines. The integration of AI allows for dynamic rule generation, real-time anomaly detection, and automated optimization, resulting in significant improvements in data quality and downstream machine learning performance. The framework demonstrates that embedding intelligence into data workflows is not only beneficial but essential for the future of large-scale ERP software systems. Future work will involve deploying the framework in live ERP environments and integrating explainable AI components for enhanced traceability and compliance.

References:

1. **Chen, M., Mao, S., & Liu, Y. (2015).** Big data: A survey. *Mobile Networks and Applications*, 19(2), 171–209.
2. **Gandomi, A., & Haider, M. (2015).** Beyond the hype: Big data concepts, methods, and analytics. *International Journal of Information Management*, 35(2), 137–144.
3. **Ibrahim, R., et al. (2016).** Cloud ERP systems: Anatomy of adoption factors & attitudes. *Journal of Enterprise Information Management*, 29(2), 304–327.
4. **Sivarajah, U., et al. (2017).** Critical analysis of Big Data challenges and analytical methods. *Journal of Business Research*, 70, 263–286.
5. **Wamba, S. F., et al. (2017).** Big data analytics and firm performance: Effects of dynamic capabilities. *Journal of Business Research*, 70, 356–365.
6. **Schelter, S., et al. (2018).** Automating large-scale data quality verification. *Proceedings of the VLDB Endowment*, 11(12), 1781–1794.
7. **Halevy, A., et al. (2018).** Machine learning for data transformation: The case of schema matching. *IEEE Data Engineering Bulletin*, 41(2), 38–47.
8. **Polyzotis, N., et al. (2018).** Data lifecycle challenges in production ML systems. *IEEE Data Engineering Bulletin*, 41(4), 5–16.
9. **Wang, R. Y., & Strong, D. M. (2019).** Beyond accuracy: What data quality means to data consumers. *Journal of Management Information Systems*, 12(4), 5–34. (Reprint of 1996 work with 2019 commentary).
10. **Almeida, F., & Calistru, C. (2019).** The main challenges of Industry 4.0. *Journal of Engineering and Technology Management*, 54, 1–9.
11. **Hellerstein, J. M., et al. (2019).** Ground: A data context service. *Proceedings of the VLDB Endowment*, 13(12), 3165–3178.
12. **Vassiliadis, P., et al. (2019).** ETL 2.0: Next-generation real-time data integration. *Information Systems Frontiers*, 21(4), 719–740.
13. **Chandrasekaran, K., et al. (2020).** AI-driven metadata management for enterprise data lakes. *Journal of Data and Information Quality*, 12(3), 1–25.

14. **Gulzar, M. A., et al. (2020).** AI for automated ETL pipeline optimization. *IEEE Transactions on Knowledge and Data Engineering*, 32(11), 2182–2196.

15. **Schellenberger, M., et al. (2020).** Reinforcement learning for dynamic resource allocation in data pipelines. *Proceedings of the ACM Symposium on Cloud Computing*, 478–491.

16. **Zhang, A., et al. (2020).** Machine learning in production: Challenges and best practices. *Communications of the ACM*, 63(4), 48–55.

17. **Krishnan, S., & Wu, E. (2021).** AlphaClean: Automatic generation of data cleaning pipelines. *Proceedings of the VLDB Endowment*, 14(12), 2885–2898.

18. **Le, Q. T., et al. (2021).** Deep learning for anomaly detection in ERP transactional data. *Expert Systems with Applications*, 178, 115016.

19. **Marz, N., & Warren, J. (2021).** *Big Data: Principles and best practices of scalable real-time data systems* (2nd ed.). Manning Publications.

20. **Ribeiro, M. T., et al. (2021).** Continuous integration for machine learning. *Journal of Systems and Software*, 180, 111026.

21. **Verma, A., et al. (2021).** AI-augmented data governance in enterprise systems. *Information Systems Management*, 38(4), 312–329.

22. **Diamantopoulos, T., et al. (2022).** Towards ML-oriented data pipelines. *IEEE Transactions on Big Data*, 8(1), 177–191.

23. **Liu, X., et al. (2022).** MetaETL: A metadata-driven framework for adaptive ETL. *Journal of Data Science*, 20(3), 421–440.

24. **Patel, P., et al. (2022).** Dynamic orchestration of data workflows using reinforcement learning. *Future Generation Computer Systems*, 134, 187–201.

25. **Zheng, A., & Casari, A. (2022).** *Feature engineering for machine learning: Principles and techniques for data scientists*. O'Reilly Media.