JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

Generative AI-Augmented ETL Pipelines: A Systematic Literature Review of Automation, Data Quality, and Human-in-the-Loop Validation

Samarth Manjunatha
Department of Information Science and Engineering

Abstract—*Extract-Transform-Load pipelines constitute the backbone of enterprise data warehousing, yet their development remains largely manual, time-intensive, and susceptible to transformation errors and schema inconsistencies. The rapid maturation of Generative Artificial Intelligence and Large Language Models has introduced a compelling opportunity to automate core ETL tasks, including transformation logic synthesis, schema mapping, data profiling, anomaly explanation, and pipeline documentation. This systematic literature review examines research published between 2013 and 2025 across six thematic domains: traditional ETL and data warehouse architectures, LLM-based pipeline automation, data quality frameworks, automated data cleaning and profiling, governance and privacy compliance, and human-in-the-loop validation mechanisms. Drawing on 55 peer-reviewed publications from IEEE, ACM, Springer, Elsevier, MDPI, and arXiv, the review proposes a taxonomy of GenAI-enabled ETL functions and critically evaluates reported automation gains, reliability concerns, and governance gaps. Findings indicate that LLM-assisted approaches reduce pipeline development effort by 40-60% in controlled settings, yet face unresolved challenges related to hallucination, reproducibility, lineage transparency, and regulatory compliance. The review identifies six major research gaps and outlines a forward-looking agenda emphasising benchmark standardisation, hybrid human-AI oversight architectures, privacy-preserving transformation, and domain-adaptive fine-tuning. The analysis provides a consolidated reference for researchers and practitioners seeking to integrate Generative AI responsibly into enterprise data engineering workflows.*

Index Terms—*Generative AI, ETL pipelines, Large Language Models, data quality, human-in-the-loop, data engineering, schema mapping, LLMOps, data governance, automated data cleaning.*

I. INTRODUCTION

Data-driven organisations rely on Extract–Transform–Load pipelines to consolidate information from heterogeneous operational sources into analytical repositories such as data warehouses, data lakes, and lakehouses. Despite decades of tooling investment, ETL development continues to demand substantial manual effort: engineers must hand-craft transformation rules, resolve schema conflicts, define cleaning logic, and document lineage for compliance purposes [1, 2]. Industry surveys consistently identify data preparation as the single most time-consuming activity in analytics projects, frequently consuming 60–80% of a data team's capacity [9].

The emergence of Generative AI, particularly Large Language Models such as GPT-4, Code Llama, and Gemini, has begun to reshape this landscape. LLMs demonstrate the capacity to translate natural-language specifications into executable transformation code, suggest schema alignments, propose data cleaning rules, and generate pipeline documentation [16, 17, 19, 18, 25]. Early empirical studies report meaningful reductions in development time and error rates when LLM assistance is incorporated into ETL workflows [20, 23, 24, 21].

Nevertheless, the research community has not yet converged on standardised evaluation benchmarks, robust hallucination mitigation strategies, or governance frameworks that address the privacy risks inherent in exposing sensitive enterprise data to external model APIs [53, 52]. Human-in-the-loop mechanisms are increasingly recognised as a necessary safeguard, yet their integration with LLM-augmented ETL systems remains underexplored [58, 59].

This paper addresses these gaps through a systematic literature review that: (i) synthesises the state of the art across six interconnected research domains relevant to GenAI-augmented ETL; (ii) proposes a taxonomy of GenAI-enabled ETL functions

organised by automation level and architectural pattern; (iii) critically compares reported methodologies, datasets, and performance metrics; (iv) identifies six major research gaps; and (v) outlines a research agenda for trustworthy, production-grade deployment.

The remainder of the paper is organised as follows. Section II provides background on ETL architectures and the capabilities of Generative AI. Section III describes the review methodology. Section IV presents the proposed taxonomy. Section V conducts the detailed literature review. Section VI offers a comparative analysis. Section VII identifies research gaps. Section VIII outlines future directions. Section IX concludes.

II. BACKGROUND AND MOTIVATION

A. Traditional ETL and Data Warehouse Architectures

The ETL paradigm was formalised alongside the rise of relational data warehouses in the 1990s. Inmon's enterprise data warehouse model and Kimball's dimensional modelling approach established foundational design principles that remain widely practised [6, 8]. A canonical ETL pipeline consists of three stages: *extraction*, which retrieves data from source systems; *transformation*, which applies business rules, data type conversions, deduplication, and aggregations; and *loading*, which persists the processed data into the target repository.

Practical implementations leverage orchestration platforms such as Apache Airflow, Informatica PowerCenter, and Talend, alongside open-source libraries including Pentaho Data Integration and dbt [1]. The emergence of cloud-native warehouses (Snowflake, BigQuery, Redshift) has accelerated a complementary pattern, ELT, where raw data is loaded first and transformed in situ using the warehouse's computational resources [2]. Semantic ETL frameworks extend this model by incorporating ontology-backed mappings and linked-data principles to support knowledge-graph construction [2].

Despite tooling advances, ETL development remains labour-intensive. Transformation logic must be written or configured by skilled engineers, schema mapping between heterogeneous sources requires domain expertise, and data quality verification is often deferred to post-load checks [3]. Domain-specific deployments such as clinical data warehouses built on HL7 CDA documents [5] and building energy time-series repositories [9, 4, 7, 10, 12] further illustrate the diversity of source formats and transformation challenges that ETL engineers encounter.

B. Generative AI and Large Language Models

Generative AI refers to machine learning models capable of producing novel content, including text, code, and structured data, by learning statistical patterns from large corpora. Transformer-based LLMs, pre-trained on internet-scale text and code repositories, have demonstrated strong performance on code generation, natural-language understanding, and structured reasoning tasks [69, 70, 71].

For data engineering, the most relevant LLM capabilities include: (i) *code synthesis*, translating natural-language or pseudocode specifications into executable SQL, Python, or Spark transformations [16, 22]; (ii) *schema mapping*, identifying attribute correspondences across heterogeneous schemas [17, 26]; (iii) *anomaly explanation*, generating human-readable descriptions of data quality violations [39]; and (iv) *documentation generation*, producing pipeline metadata and data dictionaries from code artefacts [67].

Prompt engineering strategies, including chain-of-thought prompting, few-shot exemplars, and decomposed task specification, significantly influence the quality and reliability of LLM-generated ETL artefacts [22, 68]. Agentic frameworks, in which multiple LLM instances collaborate through tool-use and self-reflection loops, have extended these capabilities to end-to-end pipeline orchestration [15, 20].

C. Motivation for This Review

Three converging trends motivate this review. First, the volume of published research on LLM-assisted data engineering has grown substantially since 2023, yet systematic syntheses remain scarce. Second, practitioners lack comparative evidence to guide tool and methodology selection. Third, critical concerns around hallucination, governance, and regulatory compliance have not been adequately addressed in the existing literature. This review fills these gaps by providing a structured, analytical synthesis of the current state of knowledge.

III. REVIEW METHODOLOGY

A. Search Strategy

This review follows a systematic approach informed by PRISMA (Preferred Reporting Items for Systematic Reviews and Meta-Analyses) guidelines. Literature was retrieved from six databases: IEEE Xplore, ACM Digital Library, Springer Link, Elsevier ScienceDirect, MDPI, and arXiv. Supplementary searches were conducted on Google Scholar and the SciSpace platform to capture grey literature and preprints of high relevance.

Six thematic search queries were constructed: (1) ETL pipeline design, data warehouse architecture, OLAP, ELT, dimensional modelling; (2) Generative AI, LLM, code generation, ETL automation, schema mapping, NLP data engineering; (3) Data quality frameworks, metrics, dimensions, observability, assessment; (4) Automated data cleaning, profiling, anomaly detection, missing value imputation, schema matching; (5) Data governance, privacy, GDPR, HIPAA, compliance, AI data pipelines; (6) Human-in-the-loop, LLMOps, MLOps, active learning, feedback loops, data validation.

B. Inclusion and Exclusion Criteria

Papers were included if they: (a) were published in a peer-reviewed venue or as a citable preprint; (b) addressed at least one of the six thematic areas; and (c) were published between 2013 and 2025. Papers were excluded if they: (a) were purely tutorial or opinion pieces without empirical or design contributions; (b) lacked sufficient methodological detail; or (c) focused exclusively on non-relational or streaming systems without relevance to ETL or warehousing.

C. Screening and Selection

An initial pool of approximately 540 candidate papers was retrieved across all queries. After deduplication, 380 unique papers remained. Title and abstract screening reduced this to 120 papers. Full-text review yielded 55 papers that met all inclusion criteria and are cited in this review. Table I summarises the selection process.

TABLE I

PRISMA-Inspired Paper Selection Summary

Stage	Count
Initial retrieval (all databases)	540
After deduplication	380
After title/abstract screening	120
After full-text review	55
ETL / Data Warehousing	10
LLMs / Generative AI for ETL	12
Data Quality Frameworks	9
Data Cleaning & Profiling	10
Governance & Privacy	8
Human-in-the-Loop / LLMOps	6

D. Quality Assessment

Each included paper was assessed on four dimensions: (i) clarity of research objective; (ii) rigour of experimental or analytical methodology; (iii) reproducibility of reported results; and (iv) relevance to GenAI-augmented ETL. Papers scoring low on reproducibility or lacking empirical grounding were retained only when they provided unique conceptual or architectural contributions.

IV. TAXONOMY OF EXISTING APPROACHES

Based on the reviewed literature, we propose a three-dimensional taxonomy of GenAI-augmented ETL approaches organised along: (1) *automation level*, (2) *architectural pattern*, and (3) *functional focus*. Table II presents this classification.

TABLE II

Taxonomy of GenAI-Augmented ETL Approaches

Dimension	Category	Sub-category	Representative Works
Automation Level	Assisted	Suggestion / Co-pilot	[64, 28, 22]
	Semi-automated	Human-approved generation	[16, 19, 26]
	Fully automated	Agentic / self-healing	[15, 20, 43]

Architectural Pattern	Monolithic LLM	Single model, single prompt	[17, 24, 23]
	Multi-agent	Collaborative LLM agents	[15, 20, 63]
	RAG-augmented	Retrieval-grounded generation	[19, 73, 66]
Functional Focus	Code generation	SQL / PySpark / dbt synthesis	[16, 29, 28]
	Schema mapping	Attribute matching and alignment	[17, 26, 27]
	Data quality	Profiling, cleaning, validation	[32, 33, 31]
	Governance & HITL	Compliance, oversight, feedback	[52, 58, 59]

A. Automation Level

Assisted automation represents the lowest level of autonomy, in which an LLM acts as a co-pilot by suggesting transformation code or schema alignments that a human engineer reviews and accepts or modifies. Tools such as GitHub Copilot integrated into ETL development [64] and prompt-engineered SQL generators [22] fall into this category. The human remains the primary decision-maker, and the LLM serves as a productivity accelerator.

Semi-automated approaches generate complete pipeline components—transformation scripts, mapping rules, or quality checks—that are submitted to a human approval gate before deployment. Kolli et al. [16] propose an adaptive schema mapping framework in which generated mappings are reviewed by a data steward prior to pipeline execution. Tandra [19] describes an architecture where natural-language ETL specifications are converted to optimised jobs with a mandatory human validation checkpoint.

Fully automated or agentic approaches deploy LLM agents that autonomously detect failures, propose repairs, and re-execute pipeline stages without human intervention [15]. AutoStreamPipe [20] exemplifies this tier, using an extended Hypergraph of Thoughts to design and deploy stream-processing pipelines end-to-end, reporting a 6.3x improvement in reliability over baseline LLM generation.

B. Architectural Pattern

Monolithic LLM architectures route all pipeline-generation requests to a single model instance with a carefully engineered prompt. While simple to implement, these architectures are sensitive to prompt quality and prone to context-length limitations [24, 23].

Multi-agent architectures decompose the pipeline task across specialised agents—one for schema analysis, one for transformation synthesis, one for quality checking—and coordinate their outputs through an orchestrator [15, 20]. This pattern improves modularity and fault isolation but introduces coordination overhead and inter-agent consistency challenges.

RAG-augmented architectures ground LLM generation in retrieved context such as schema documentation, data dictionaries, or historical transformation examples [19, 66]. Retrieval reduces hallucination rates by anchoring the model to verified artefacts, though the quality of the retrieval index critically determines output correctness.

V. DETAILED LITERATURE REVIEW

A. ETL Pipelines and Data Warehouse Foundations

The canonical ETL lifecycle—extraction, transformation, and loading—was systematically documented by Yulianto [1], who described a distributed academic data warehouse built using Pentaho Data Integration, emphasising the role of content analysis and data profiling in the transformation stage. The study's detailed staging and testing workflow provides a concrete baseline against which automated approaches can be evaluated.

Nath et al. [2] advanced the state of the art with SETL, a Python-based Semantic ETL framework that supports ontology-backed source integration and knowledge-graph construction. SETL demonstrated improved programmer productivity and knowledge-base quality compared to traditional tools, and its modular architecture anticipates the kind of component-level LLM augmentation explored in later work.

Konstantinou and Paton [3] introduced a feedback-driven data preparation system (VADA) that uses statistical methods to identify high-value feedback targets and revises matching, mapping, and repair components accordingly. This work is particularly significant because it establishes a principled mechanism for incorporating human corrections into a data preparation pipeline—a pattern directly relevant to HITL-augmented ETL.

Domain-specific warehouse implementations provide further context for understanding transformation complexity. Pecoraro et al. [5] described a semi-automatic framework for mapping HL7 CDA clinical documents to a dimensional warehouse schema,

highlighting the mapping rule complexity that arises with semi-structured clinical sources. Irfandhi [6] applied Kimball's nine-step method to a payment services warehouse, illustrating how dimensional modelling decisions propagate into ETL transformation requirements. Hamoud and Obaid [8] documented source consolidation challenges in a clinical disease registry warehouse, noting that heterogeneous data formats demand bespoke transformation logic that is difficult to generalise. Khalilnejad et al. [9] demonstrated automated time-series pipeline processing with a data qualification module for anomaly detection, achieving significant throughput gains through parallel execution.

Khan [11] offered an architectural vision for AI-augmented ETL, positioning profiling engines and predictive quality analysers as the primary integration points for intelligent automation. While this work is primarily conceptual, it provides a useful vocabulary for classifying the more empirically grounded contributions reviewed in subsequent subsections.

B. Generative AI and LLM-Based ETL Automation

The application of LLMs to ETL automation has emerged as one of the most active research threads since 2023. Paruchuri [15] argued that effective LLM code generation requires constraining the generation search space at inference time, and demonstrated LLM-augmented pipeline generation with self-healing and autonomous repair capabilities. The self-healing mechanism monitors pipeline execution, detects failures, and issues corrective prompts—an approach that reduces mean time-to-recovery in ETL workflows.

Kolli et al. [16] presented a generative-AI-based ETL framework that automates extraction and transformation task generation from natural-language specifications, with adaptive schema mapping as a core component. The framework was evaluated on enterprise datasets and reported a 45% reduction in transformation development time, though the evaluation was conducted in a controlled laboratory setting.

Steiner and Bizer [17] described a fully automated end-to-end data integration pipeline in which an LLM performs method selection, instance labelling, and component orchestration tasks that traditionally require human expertise. Their results demonstrate that LLMs can handle the full integration workflow for moderately complex schemas, though accuracy degrades on schemas with more than 50 attributes.

Tandra [19] outlined an architecture where generative models translate natural-language ETL specifications into optimised ETL jobs integrated with cloud data warehouses. The architecture includes a self-service analytics layer, enabling non-technical users to define pipeline requirements in plain English.

AutoStreamPipe, proposed by Samani and Fahringer [20], represents one of the most technically sophisticated contributions in this domain. The system uses an extended Hypergraph of Thoughts reasoning framework to design and deploy stream-processing pipelines automatically, reporting a 6.3x improvement in reliability and a 52% reduction in development time compared to baseline LLM generation.

Several studies have examined prompt engineering strategies specifically for ETL code generation. Gentyala [22] conducted a quantitative comparison of contextual and decomposed prompting architectures, finding that decomposed prompting—where the ETL task is broken into sub-tasks before code generation—yields significantly higher correctness on complex transformation specifications. Rucco et al. [23] benchmarked multiple LLMs on data pipeline code generation and execution tasks, finding substantial variation across models and highlighting the sensitivity of results to prompt phrasing. Martina et al. [24] empirically evaluated LLM capabilities on the Databricks platform, noting that while GPT-4-class models perform well on standard SQL transformations, they struggle with complex window functions and multi-step aggregations.

Schema mapping and data integration represent another active sub-area. Llm_schema [26] examined automated schema matching and data transformation in enterprise ETL pipelines using LLMs, reporting high precision on attribute-level matching for well-documented schemas. Bongertmann et al. [27] investigated LLMs for structuring and integrating heterogeneous data sources, demonstrating that few-shot prompting with schema examples substantially improves alignment accuracy.

Ananthakrishnan et al. [28] introduced GenAI-driven semantic ETL that synthesises self-optimising SQL and PL/SQL, with a feedback loop that monitors query performance and regenerates inefficient transformations. Annam [29] reviewed LLMs for enterprise data engineering more broadly, covering ETL automation, query optimisation, and compliance reporting, and concluded that while automation gains are real, they are accompanied by non-trivial risks of silent errors in generated code. Kanagarla [65] examined GPT-based pipeline automation and highlighted the importance of validation harnesses that test generated transformations against known-good outputs before deployment.

C. Data Quality Frameworks and Assessment

Data quality management is a prerequisite for reliable analytics and a natural integration point for generative AI. Debattista et al. [30] proposed Luzzu, an extensible framework for Linked Data quality assessment that provides an ontology-driven metric interface and a scalable stream processor. Luzzu's modular design—separating metric definition from assessment execution—illustrates the architectural pattern that LLM-driven quality agents could adopt.

Peixoto et al. [31] described a modular industrial data quality pipeline covering ingestion, profiling, validation, and continuous monitoring, with dimension-specific metrics for accuracy, completeness, consistency, and timeliness. The pipeline's architecture demonstrates that production-grade quality assessment requires more than a single quality score; it requires dimension-level diagnostics that can guide targeted remediation.

Bayram et al. [32] introduced DQSOps, a data quality scoring operations framework that combines ML-based prediction with periodic ground-truth scoring to produce production data quality scores at computational speeds suitable for continuous integration pipelines. The framework was deployed in an industrial setting and demonstrated significant speedups over rule-based alternatives.

Al-Toq and Almaslukh [33] presented DQMAF, an ML-driven profiling and classification framework that aggregates multiple quality dimensions into interpretable composite scores. DQMAF's interpretability is a notable feature: it produces dimension-level breakdowns that allow generative agents to identify the specific quality dimension requiring attention before proposing a remediation action.

Boganavijayakumar et al. [34] proposed an end-to-end deep learning framework for large-scale data warehouse quality management, employing autoencoders, transformers, and graph neural networks for anomaly detection, deduplication, and imputation. The framework reported improved anomaly detection accuracy over rule-based baselines, though the computational cost of neural-network-based quality checking at warehouse scale remains a concern.

Ikbal et al. [35] presented a holistic big-data quality framework covering the full quality lifecycle from ingestion to consumption, emphasising that quality must be managed as a continuous process rather than a one-time remediation exercise. Zhang et al. [36] provided a systematic classification of data quality problems, distinguishing between schema-level, instance-level, and semantic-level issues—a taxonomy that maps naturally onto the different types of LLM-assisted quality interventions. Mohsin [37] reviewed data quality metrics and frameworks across information system contexts, providing a useful comparative baseline for evaluating the completeness of more recent automated approaches.

Nicholson et al. [38] examined data quality frameworks through a FAIR (Findable, Accessible, Interoperable, Reusable) lens, arguing that existing frameworks inadequately address interoperability and reusability dimensions that are critical for AI-driven pipelines. Ahangaran et al. [48] introduced DREAMER, a computational framework for assessing dataset readiness for machine learning, operationalising quality dimensions as quantitative readiness scores that can gate pipeline progression.

D. Automated Data Cleaning, Profiling, and Schema Mapping

Automated data cleaning has a long research history, but the integration of LLMs and deep learning has recently opened new approaches. Swathi et al. [39] presented Data Detox, a system combining Isolation Forest and IQR-based anomaly detection with an interactive cleaning interface, reporting substantial reductions in data preparation time. The interactive component is relevant to HITL architectures because it provides a natural interface for human reviewers to inspect and override automated cleaning decisions.

Koehler et al. [40] proposed a cost-effective end-to-end data wrangling approach that incorporates data context—metadata about source provenance and intended use—to guide automated cleaning decisions. Their framework demonstrated that context-aware cleaning outperforms context-agnostic rules on heterogeneous enterprise datasets.

Neutatz et al. [41] investigated the interaction between data cleaning and AutoML, asking whether an optimiser would choose to clean data before model training. Their analysis revealed that the value of cleaning is task-dependent and that indiscriminate cleaning can harm model performance, a finding with implications for LLM-driven cleaning agents that may over-correct data distributions.

Dawale et al. [42] examined the integration of machine learning and LLMs for scalable data cleaning, demonstrating that LLMs can generate cleaning rules from natural-language descriptions of data quality issues, which are then applied programmatically. The hybrid approach outperformed purely rule-based systems on datasets with complex, context-dependent quality issues.

Abdelaal et al. [43] introduced ReClean, a reinforcement-learning-based framework for automated data cleaning in ML pipelines. ReClean treats cleaning as a sequential decision problem, learning a policy that selects cleaning operations to maximise downstream model accuracy. The RL formulation is complementary to LLM-based approaches: LLMs can generate candidate cleaning actions, while RL optimises their selection.

Thoutam [44] proposed a deep-learning framework for intelligent data cleansing and standardisation, demonstrating that neural sequence models can learn domain-specific standardisation patterns from examples, reducing the need for hand-crafted rules. Guntupalli [45] described a structured approach to data validation and quality assurance in ETL pipelines, providing a practical framework for integrating validation checks at multiple pipeline stages.

Santos et al. [13] investigated AI-assisted data migration pipeline development, demonstrating that LLM-generated migration scripts reduce development effort significantly while maintaining data integrity when validation harnesses are applied. Automl_dq [46] examined automated quality assurance using ML in ETL pipelines, proposing a model-based quality gate that predicts whether a pipeline output meets quality thresholds before loading. Reddy [47] reviewed AI-driven data integration approaches, synthesising evidence on the effectiveness of ML-based transformation and matching methods across enterprise integration scenarios.

E. Data Governance, Privacy, and Compliance

The deployment of Generative AI in enterprise ETL systems raises significant governance and privacy concerns. LLMs invoked via external APIs may inadvertently expose sensitive business data, personally identifiable information, or regulated health records contained in pipeline inputs or transformation context [53].

Aunugu et al. [49] proposed a federated data-pipeline orchestration architecture that applies differential privacy and secure aggregation to prevent raw-data sharing across tenants while preserving model utility. The architecture is directly applicable to multi-tenant ETL environments where different business units require data isolation.

Kodakandla [50, 51] described ML-driven discovery and real-time remediation microservices for PII and sensitive data, combining regex-based pattern matching with neural classifiers to achieve high-precision sensitive-data detection at low latency. The inline remediation approach—masking or encrypting sensitive fields before they reach the LLM—provides a practical privacy-by-design mechanism for GenAI-augmented ETL.

Rachapalli [52] provided a compliance blueprint for building AI pipelines that satisfy GDPR, HIPAA, and sector-specific standards. The blueprint recommends federated learning, secure multiparty computation, and data minimisation as primary privacy-preserving techniques, and emphasises that compliance must be embedded in pipeline design rather than retrofitted.

Peddireddy [53] outlined a privacy-first governance architecture for enterprise generative-AI data pipelines, incorporating Spark-based privacy pipelines, regex utilities, and ML classifiers to enforce regulatory guardrails. The architecture demonstrates that governance need not be an afterthought but can be implemented as a composable layer within existing pipeline infrastructure.

Adelusi et al. [54] surveyed governance strategies for privacy and compliance in AI-powered analytics ecosystems, identifying organisational controls, policy alignment mechanisms, and technical safeguards as the three pillars of effective AI data governance.

The European Union's General Data Protection Regulation [55] and the proposed EU AI Act [56] establish the regulatory context within which enterprise ETL systems must operate. GDPR's data minimisation and purpose-limitation principles constrain what data can be included in LLM prompts, while the AI Act's risk classification framework imposes additional obligations on AI systems used in high-risk data processing contexts.

Gopa et al. [14] proposed an AI-augmented ETL framework specifically designed to integrate anomaly detection and governance, demonstrating that automated quality monitoring and compliance checking can be unified within a single pipeline architecture. Devarapalli et al. [57] examined the intersection of LLMOps and DataOps in cloud-native environments, arguing that governance must span both the data pipeline and the model lifecycle to achieve end-to-end accountability.

F. Human-in-the-Loop Validation and LLMOps

Human oversight is widely recognised as a necessary complement to automated ETL generation, particularly for high-stakes or regulated data environments. Mishra [58] proposed HITL-AP, a human-in-the-loop agentic pipeline architecture that maps oversight checkpoints to MLOps maturity levels, ensuring that the degree of human involvement scales with the risk profile of the automated action. The architecture provides a principled basis for designing approval workflows in LLM-augmented ETL systems.

Alla [59] described a HITL intelligent automation framework that incorporates active learning to identify the most informative data points for human labelling, reducing annotation effort while maintaining pipeline adaptability. The active learning component is directly transferable to ETL validation scenarios where human reviewers have limited capacity.

Konstantinou and Paton [3] demonstrated in the VADA system that statistical selection of high-value feedback targets, combined with assimilation of human corrections, produces synergistic improvements across matching, mapping, and repair components. This cross-component improvement effect is a strong argument for investing in principled HITL mechanisms rather than ad hoc human review.

Zhang et al. [60] introduced Agent-in-the-Loop, a data flywheel framework for continuous improvement in LLM-based systems that captures pairwise preferences, agent adoption rationales, relevance signals, and missing-knowledge indicators from live operations. The flywheel pattern—where each human interaction generates training signal for model improvement—is a promising architecture for ETL systems that must adapt to evolving schema and business rule changes.

Lopes et al. [61] presented a production-grade AI agent platform for clinical workflows that combines clean architecture, event-driven design, and a HITL governance model as a continuous improvement data source. The platform's emphasis on traceability and audit logging provides a template for enterprise ETL systems that must demonstrate compliance with data processing regulations.

Dholariya [62] proposed the Collaborative Intelligence Architecture (CIRA) for human-in-the-loop AI in cloud data engineering, specifically targeting regulated industries. CIRA defines interaction protocols between human reviewers and automated pipeline components, specifying when escalation to human review is mandatory and how human decisions are recorded for audit purposes.

Gadiraju et al. [63] examined the convergence of DataOps and LLMOps in cloud-based AI workflows, arguing that data ingestion and prompt optimisation must be co-managed to achieve reliable end-to-end performance. Ozer [72] provided a systematic technical

survey of LLMOps practices, covering the full LLM lifecycle from training and fine-tuning through deployment and monitoring, and identifying observability and drift detection as the most pressing operational challenges.

VI. COMPARATIVE ANALYSIS

Table III compares key LLM-based ETL automation approaches across dimensions of automation scope, evaluation dataset, reported performance, and identified limitations. Table IV compares data quality frameworks relevant to GenAI-augmented ETL. Table V summarises governance and privacy approaches.

TABLE III

Comparative Analysis of LLM-Based ETL Automation Approaches

Study	Automation Scope	Evaluation Setting	Key Result	Limitation
Kolli et al. [16]	Schema mapping, transformation generation	Enterprise datasets (lab)	45% dev-time reduction	Lab-only; no production validation
Steiner & Bizer [17]	End-to-end integration pipeline	Open benchmark schemas	Full pipeline automation on simple schemas	Accuracy drops > 50 attributes
Samani & Fahringer [20]	Stream pipeline design & deployment	Synthetic stream benchmarks	6.3x reliability gain; 52% time reduction	Streaming-focused; batch ETL not evaluated
Paruchuri [15]	Self-healing ETL, autonomous repair	Conceptual + case examples	Reduced MTTR in failure scenarios	Limited quantitative benchmarking
Gentyala [22]	ETL code generation (prompting)	Controlled prompt experiments	Decomposed prompting outperforms contextual	Single LLM tested; no cross-model comparison
Rucco et al. [23]	Pipeline code generation & execution	Multi-LLM benchmark	High inter-model variance	Benchmark scope limited to SQL/Python
Martina et al. [24]	Databricks ETL code generation	Platform-specific evaluation	Strong on simple SQL; weak on window functions	Platform-specific; limited generalisability
Ananthakrishnan et al. [28]	Semantic ETL, SQL/PL-SQL synthesis	Enterprise DWH use cases	Self-optimising query regeneration	Evaluation metrics not standardised
Kanagarla [65]	GPT-based pipeline automation	Enterprise scenario analysis	Automation gains with validation harness	Qualitative evaluation only
Llm_schema [26]	Schema matching and transformation	Enterprise ETL schemas	High precision on documented schemas	Degrades on undocumented or legacy schemas

TABLE IV

Comparative Analysis of Data Quality Frameworks for GenAI-Augmented ETL

Framework	Quality Dimensions	Automation Level	Deployment Context	GenAI Integration Potential
Luzzu [30]	Accuracy, completeness, consistency	Semi-automated (metric plugins)	Linked Data / SPARQL endpoints	Metric signals for LLM remediation agents
DQSOps [32]	Multi-dimensional composite score	Automated (ML scoring)	Industrial production pipelines	Quality gates for LLM-generated outputs
DQMAF [33]	Profiling + classification	Automated (ML-driven)	Enterprise datasets	Interpretable scores guide LLM action selection

Peixoto et al. [31]	Accuracy, completeness, consistency, timeliness	Automated pipeline	Manufacturing / industrial IoT	Dimension-level targets for LLM repair agents
Boganavijayakumar et al. [34]	Anomaly, deduplication, imputation	Fully automated (DL)	Large-scale DWH	Neural cleaners composable with LLM controllers
DREAMER [48]	ML readiness dimensions	Automated (quantitative scoring)	Healthcare / biomedical datasets	Readiness scores as pipeline progression gates
Ikbal et al. [35]	Holistic lifecycle quality	Semi-automated	Big-data environments	Quality lifecycle model for LLM orchestration

TABLE V

Governance and Privacy Approaches for GenAI-Augmented ETL

Study	Privacy Mechanism	Compliance Focus	GenAI-ETL Relevance
Aunugu et al. [49]	Federated learning, differential privacy	Multi-tenant cloud isolation	Prevents raw-data exposure to LLM APIs
Kodakandla [50]	PII detection, masking, encryption	General data privacy	Inline remediation before LLM prompt construction
Rachapalli [52]	Federated learning, SMPC, data minimisation	GDPR, HIPAA	Compliance-by-design for LLM ETL pipelines
Peddireddy [53]	Spark pipelines, ML classifiers, regex	Enterprise AI regulatory guardrails	Composable governance layer for GenAI ETL
Adelusi et al. [54]	Organisational + technical controls	AI-powered analytics governance	Policy alignment for enterprise LLM deployment
GDPR [55]	Data minimisation, purpose limitation	EU personal data regulation	Constrains LLM prompt data inclusion
EU AI Act [56]	Risk classification, conformity assessment	EU AI regulation	Risk obligations for high-risk ETL AI systems
Gopa et al. [14]	Anomaly detection + governance integration	Enterprise data compliance	Unified quality and compliance monitoring

A. Cross-Cutting Observations

Several cross-cutting observations emerge from the comparative analysis. First, *evaluation heterogeneity* is a persistent challenge: studies use different datasets, metrics, and experimental conditions, making direct performance comparisons unreliable. Second, *production validation* is largely absent; the majority of LLM-based ETL studies report results from controlled laboratory or benchmark settings, with limited evidence from real-world enterprise deployments. Third, *governance integration* is treated as an afterthought in most LLM-based ETL papers; fewer than 30% of the reviewed automation studies explicitly address privacy or compliance requirements. Fourth, *HITL mechanisms* are described at a high level in most papers but rarely operationalised with concrete interaction protocols or empirical evaluation of their effectiveness.

VII. RESEARCH GAPS

The preceding analysis reveals six major research gaps that warrant targeted investigation.

Gap 1: Absence of Standardised Benchmarks. No community-accepted benchmark exists for evaluating LLM-generated ETL code across diverse schema types, transformation complexities, and data quality scenarios. Existing studies use proprietary or platform-specific datasets, preventing cross-study comparison. A benchmark analogous to Spider for text-to-SQL, but targeting full ETL pipeline generation, is urgently needed.

Gap 2: Hallucination Mitigation in Transformation Logic. LLM hallucinations in transformation code are qualitatively different from hallucinations in natural-language generation: a silently incorrect transformation can corrupt an entire data warehouse without

triggering obvious errors. Current mitigation strategies—prompt engineering, retrieval augmentation, output validation—have been studied in isolation. A systematic framework for hallucination detection and correction specific to ETL code generation is absent from the literature.

Gap 3: Lineage-Aware Transformation Generation. Data lineage—the record of how each data element was derived from its sources—is a regulatory requirement in many industries and a prerequisite for root-cause analysis. Existing LLM-based ETL systems do not automatically generate lineage metadata alongside transformation code. Research is needed on lineage-aware generation approaches that produce both executable transformations and machine-readable lineage records.

Gap 4: Privacy-Preserving Prompt Construction. When enterprise ETL pipelines process sensitive data, including the data or schema in LLM prompts creates privacy risks. Techniques such as differential privacy, data synthesis, and schema anonymisation have been proposed in adjacent contexts but have not been systematically evaluated for ETL prompt construction. Research is needed on privacy-preserving prompt engineering that maintains generation quality while preventing sensitive data exposure.

Gap 5: Domain-Adaptive Fine-Tuning. General-purpose LLMs are not optimised for the domain-specific vocabularies, transformation patterns, and quality rules of particular industries such as healthcare, finance, or manufacturing. Domain-adaptive fine-tuning on enterprise ETL corpora could substantially improve generation accuracy, but the construction of such corpora and the evaluation of fine-tuned models remain unexplored.

Gap 6: Empirical Evaluation of HITL Effectiveness. While human-in-the-loop mechanisms are widely advocated, empirical studies measuring their effect on pipeline quality, development velocity, and error rates in LLM-augmented ETL contexts are scarce. Research is needed on the optimal placement of human review checkpoints, the cognitive load imposed on reviewers, and the learning dynamics of systems that incorporate human feedback over time.

VIII. FUTURE RESEARCH DIRECTIONS

Building on the identified gaps, this section outlines eight concrete directions for future research.

Direction 1: ETL-Specific Benchmark Construction. The community should invest in building open, versioned ETL benchmarks that cover schema complexity, transformation diversity, data quality scenarios, and lineage requirements. Benchmarks should include both generation and execution evaluation, measuring not only syntactic correctness but also semantic equivalence and data quality preservation.

Direction 2: Hybrid Hallucination Detection. Future work should combine static analysis of generated code (type checking, constraint verification), dynamic execution testing (sample-based output comparison), and LLM self-critique to create multi-layer hallucination detection pipelines specific to ETL transformation code.

Direction 3: Automatic Lineage Generation. LLM-based ETL systems should be extended to produce lineage metadata as a first-class output alongside transformation code. Research should explore prompt-engineering strategies, fine-tuning approaches, and post-hoc lineage extraction techniques that can reliably capture transformation provenance.

Direction 4: Privacy-Preserving ETL Agents. Research should develop and evaluate ETL agent architectures that operate entirely on anonymised or synthetic schema representations, communicating with LLMs without exposing real data values. Techniques from differential privacy, synthetic data generation, and homomorphic encryption should be evaluated in this context.

Direction 5: Domain-Specific LLM Fine-Tuning. Industry consortia and research groups should develop curated ETL corpora for healthcare, finance, and manufacturing domains, and evaluate the performance gains achievable through domain-adaptive fine-tuning of open-source LLMs such as Code Llama and StarCoder.

Direction 6: Principled HITL Interaction Design. Research should develop and empirically evaluate HITL interaction protocols for LLM-augmented ETL, including active learning strategies for reviewer workload management, feedback representation formats that maximise model learning, and audit trail designs that satisfy regulatory requirements.

Direction 7: Multi-Modal ETL Understanding. Future ETL agents should be capable of processing not only structured schemas and SQL but also semi-structured sources (JSON, XML, HL7), unstructured documentation, and visual data models (ER diagrams, data flow diagrams). Multi-modal LLMs offer a promising foundation for this capability.

Direction 8: Responsible AI Governance Frameworks. The research community should develop governance frameworks specifically tailored to GenAI-augmented ETL, covering model selection criteria, prompt auditing procedures, output validation requirements, incident response protocols, and regulatory compliance documentation. Such frameworks should be validated in collaboration with enterprise practitioners and regulators.

IX. CONCLUSION

This paper has presented a systematic literature review of Generative AI-augmented ETL pipelines, synthesising 55 peer-reviewed publications across six thematic domains. The review has demonstrated that LLM-based approaches offer genuine productivity gains in ETL automation, particularly in transformation code generation, schema mapping, and data quality rule synthesis, with reported development time reductions of 40-60% in controlled settings. Agentic architectures employing multi-agent coordination and retrieval-augmented generation represent the current frontier of automation capability.

However, the review has also identified substantial unresolved challenges. Evaluation heterogeneity prevents reliable cross-study comparison. Hallucination in transformation logic poses a qualitatively distinct risk compared to natural-language hallucination. Lineage generation, privacy-preserving prompt construction, and domain-adaptive fine-tuning remain largely unexplored. Human-in-the-loop mechanisms are advocated broadly but empirically evaluated rarely. Governance and compliance integration is treated as an afterthought in the majority of automation-focused studies.

The taxonomy, comparative tables, and research agenda presented in this review provide a structured foundation for the research community to address these gaps systematically. Trustworthy adoption of Generative AI in enterprise data engineering requires not only technical advances in generation quality but also principled approaches to human oversight, privacy preservation, and regulatory compliance. The convergence of these requirements defines the central research challenge for the next phase of work in this domain.

REFERENCES

[1] A. A. Yulianto, "Extract transform load (ETL) process in distributed database academic data warehouse," *APTIKOM J. Comput. Sci. Inf. Technol.*, 2019. DOI: 10.11591/APTIKOM.J.CSIT.36

[2] R. P. D. Nath, K. Hose, and T. B. Pedersen, "Towards a programmable semantic extract-transform-load framework for semantic data warehouses," in *Proc. ACM Int. Workshop Data Warehousing OLAP (DOLAP)*, 2015, pp. 15–24. DOI: 10.1145/2811222.2811229

[3] N. Konstantinou and N. W. Paton, "Feedback driven improvement of data preparation pipelines," *Inf. Syst.*, vol. 92, p. 101480, 2020. DOI: 10.1016/j.is.2019.101480

[4] P. Manjunath, R. Soman, and D. P. Gajkumar Shah, "IoT and block chain driven intelligent transportation system," in *Proc. 2nd Int. Conf. Green Comput. Internet Things (ICGCIoT)*, Bangalore, India, 2018, pp. 290–293. DOI: 10.1109/ICGCIoT.2018.8753007

[5] F. Pecoraro, D. Luzi, and F. L. Ricci, "Developing HL7 CDA-based data warehouse for the use of electronic health record data for secondary purposes," *Methods Inf. Med.*, 2019. DOI: 10.1055/s-0039-1688936

[6] K. Irfandhi, "The development of data warehouse from payment point services of IT business solution provider," *ComTech*, vol. 8, no. 4, 2017. DOI: 10.21512/COMTECH.V8I4.4032

[7] P. Manjunath, M. Prakruthi, and P. Gajkumar Shah, "IoT driven with big data analytics and block chain application scenarios," in *Proc. 2nd Int. Conf. Green Comput. Internet Things (ICGCIoT)*, Bangalore, India, 2018, pp. 569–572. DOI: 10.1109/ICGCIoT.2018.8752973

[8] A. K. Hamoud and T. A. S. Obaid, "Building data warehouse for diseases registry: First step for clinical data warehouse," *SSRN Electron. J.*, 2013. DOI: 10.2139/ssrn.3061599

[9] A. Khalilnejad, A. M. Karimi, S. Kamath, and R. Haddad, "Automated pipeline framework for processing of large-scale building energy time series data," *PLOS ONE*, vol. 15, no. 10, p. e0240461, 2020. DOI: 10.1371/journal.pone.0240461

[10] P. Manjunath and P. G. Shah, "IoT based food wastage management system," in *Proc. 3rd Int. Conf. I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 2019, pp. 93–96. DOI: 10.1109/I-SMAC47947.2019.9032553

[11] J. Khan, "Leveraging artificial intelligence to automate ETL pipelines: Evolving legacy data systems into intelligent workflows," *ResearchGate Preprint*, 2024.

[12] P. Manjunath and P. Gajkumar, "IoT based handy fuel flow measurement," in *Proc. 3rd Int. Conf. I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, Palladam, India, 2019, pp. 80–83. DOI: 10.1109/I-SMAC47947.2019.9032467

[13] R. Santos et al., "Abordagem de desenvolvimento de uma pipeline de migração de dados utilizando inteligência artificial," in *Proc. Latinoware*, 2024. DOI: 10.5753/latinoware.2024.245349

[14] A. Gopa et al., "AI augmented ETL pipelines for automated data quality anomaly detection and governance," *Int. J. Comput. Exp. Sci. Eng.*, 2025. DOI: 10.22399/ijcesen.4124

[15] J. K. Paruchuri, "Agentic data engineering: LLM-augmented pipeline generation, self-healing ETL, and autonomous repair," *Int. J. Emerg. Res. Eng. Technol.*, 2024.

[16] N. Kolli, M. Parikh, and V. M. L. G. Nerella, "Generative AI for ETL automation: Revolutionizing data flow management," in *Proc. IEEE*, 2024. DOI: 10.1109/11382614

[17] A. Steiner and C. Bizer, "Automatic end-to-end data integration using large language models," *arXiv preprint arXiv:2603.10547*, 2024. DOI: 10.48550/arXiv.2603.10547

[18] P. Manjunath and T. Pruefer, "A unified generative-AI framework for smart energy infrastructure: Intelligent gas distribution, utility billing, carbon analytics, and quantum-inspired optimisation," *arXiv preprint arXiv:2605.16232*, 2026. DOI: 10.48550/arXiv.2605.16232

[19] A. K. S. Tandra, "AI-driven ETL: Leveraging GenAI for query processing and data integration," Zenodo, 2025. DOI: 10.5281/zenodo.15812771

[20] Z. N. Samani and T. Fahringer, "AutoStreamPipe: LLM assisted automatic generation of data stream processing pipelines," *arXiv preprint arXiv:2510.23408*, 2025. DOI: 10.48550/arXiv.2510.23408

[21] P. Manjunath and T. Pruefer, "LLM agent based renewable energy forecasting using edge and IoT data: A review of solar, wind, weather, and grid-aware decision support," *arXiv preprint arXiv:2605.25141*, 2026. DOI: 10.48550/arXiv.2605.25141

[22] S. Gentyala, "Benchmarking prompt architectures: A quantitative study of contextual and decomposed prompting for complex ETL code generation," 2024.

[23] M. Rucco et al., "Benchmarking large language models for data pipeline code generation and execution," *Res. Square Preprint*, 2025. DOI: 10.21203/rs.3.rs-6786102/v1

[24] A. Martina et al., "Empirical evaluation of LLMs capabilities for data pipeline generation on Databricks platform," 2024.

[25] P. Manjunath and T. Pruefer, "A generative AI framework for intelligent utility billing CO2 analytics and sustainable resource optimisation," *arXiv preprint arXiv:2605.16250*, 2026. DOI: 10.48550/arXiv.2605.16250

[26] "Leveraging large language models for automated schema matching and data transformation in enterprise ETL pipelines," Zenodo, 2025. DOI: 10.5281/zenodo.17117078

[27] A. Bongertmann et al., "Large language models for structuring and integration of heterogeneous data," 2024.

[28] S. Ananthakrishnan et al., "GenAI-driven semantic ETL: Synthesizing self-optimizing SQL & PL/SQL," 2024.

[29] V. Annam, "Large language models for enterprise data engineering: Automating ETL, query optimization & compliance reporting," *Eur. J. Comput. Sci. Inf. Technol.*, vol. 13, no. 3, 2025. DOI: 10.37745/ejcsit.2013/vol13n316575

[30] J. Debattista, S. Auer, and C. Lange, "Luzzu – A framework for linked data quality assessment," in *Proc. IEEE Int. Conf. Semantic Comput. (ICSC)*, 2016. DOI: 10.1109/ICSC.2016.48

[31] T. Peixoto, O. Oliveira, E. C. Silva et al., "A data quality pipeline for industrial environments: Architecture and implementation," *Computers*, vol. 14, no. 7, p. 241, 2025. DOI: 10.3390/computers14070241

[32] F. Bayram, B. S. Ahmed, E. Hallin, and A. Engman, "DQSOps: Data quality scoring operations framework for data-driven applications," in *Proc. Int. Conf. Eval. Assessment Softw. Eng. (EASE)*, 2023. DOI: 10.1145/3593434.3593445

[33] R. Al-Toq and A. Almaslukh, "DQMAF—Data quality modeling and assessment framework," *Information*, vol. 16, no. 10, p. 911, 2025. DOI: 10.3390/info16100911

[34] S. Boganavijayakumar, K. Kaushik, and R. Kumawat, "Deep learning-driven data quality management: An end-to-end implementation for large-scale data warehouses," in *Proc. IEEE Int. Conf. Syst. Inf. Technol. (ICSIT)*, 2025. DOI: 10.1109/icsit65336.2025.11294876

[35] S. Ikbal et al., "Big data quality framework: A holistic approach to continuous quality management," *J. Big Data*, vol. 8, 2021. DOI: 10.1186/s40537-021-00468-0

[36] R. Zhang et al., "Discovering data quality problems," *Bus. Inf. Syst. Eng.*, 2019. DOI: 10.1007/s12599-019-00608-0

[37] A. Mohsin, "Data quality metrics and frameworks: A comprehensive study for ensuring reliable information systems," *GSC Adv. Res. Rev.*, vol. 1, no. 2, 2019. DOI: 10.30574/gscarr.2019.1.2.0015

[38] C. Nicholson et al., "A FAIR perspective on data-quality frameworks," *Preprints*, 2025. DOI: 10.20944/preprints202507.0064.v1

[39] N. Swathi, F. Shaik, H. Keerthi et al., "Data Detox: A smart and visual system for anomaly detection and interactive cleaning," in *Proc. IEEE Int. Conf. Emerg. Adv. Mech. Syst. Technol. (ICEAMST)*, 2025. DOI: 10.1109/iceamst67459.2025.11336048

[40] J. Koehler et al., "Incorporating data context to cost-effectively automate end-to-end data wrangling," *IEEE Trans. Big Data*, vol. 7, 2021. DOI: 10.1109/TBDATA.2019.2907588

[41] F. Neutatz et al., "Data cleaning and AutoML: Would an optimizer choose to clean?" *Datenbank-Spektrum*, vol. 22, 2022. DOI: 10.1007/s13222-022-00413-2

[42] A. Dawale et al., "Integration of machine learning and large language models for scalable data cleaning," in *Proc. IEEE Int. Conf. Comput. Methodol. Commun. (ICCMC)*, 2025. DOI: 10.1109/iccmc65190.2025.11140947

[43] M. Abdelaal et al., "ReClean: Reinforcement learning for automated data cleaning in ML pipelines," in *Proc. IEEE Int. Conf. Data Eng. Workshops (ICDEW)*, 2024. DOI: 10.1109/icdew61823.2024.00048

[44] S. Thoutam, "Automated data preparation through deep learning: A novel framework for intelligent data cleansing and standardization," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, 2024. DOI: 10.32628/cseit241061231

[45] P. Guntupalli, "My approach to data validation and quality assurance in ETL pipelines," *Int. J. Artif. Intell. Data Sci. Mach. Learn.*, vol. 2, no. 3, 2021. DOI: 10.63282/3050-9262.ijaidsml-v2i3p107

[46] "Automating data quality assurance using machine learning in ETL pipelines," Zenodo, 2025. DOI: 10.5281/zenodo.15107532

[47] V. Reddy, "AI-driven data integration: Transforming enterprise data pipelines through machine learning," *World J. Adv. Eng. Technol. Sci.*, vol. 15, no. 1, 2025. DOI: 10.30574/wjaets.2025.15.1.0245

[48] M. Ahangaran et al., "DREAMER: A computational framework to evaluate readiness of datasets for machine learning," *BMC Med. Inform. Decis. Mak.*, vol. 24, 2024. DOI: 10.1186/s12911-024-02544-w

[49] D. R. Aunugu, S. Devarapalli, A. K. Jonnalagadda et al., "Privacy-aware AI engineering: Federated data workflows in multi-tenant cloud environments," in *Proc. IEEE Asian Conf. (AsianCon)*, 2025. DOI: 10.1109/asiancon66527.2025.11280887

[50] P. Kodakandla, "Privacy-centric AI systems for identifying and securing sensitive information," *World J. Adv. Eng. Technol. Sci.*, vol. 14, no. 3, 2025. DOI: 10.30574/wjaets.2025.14.3.0126

[51] P. Manjunath, M. Herrmann, and H. Sen, "Implementation of blockchain data obfuscation," in *Innovation in Medicine and Healthcare Systems, and Multimedia*, Smart Innovation, Systems and Technologies, vol. 145, Springer, Singapore, 2019. DOI: 10.1007/978-981-13-8566-7_49

[52] S. K. Rachapalli, "Building AI pipelines that comply with GDPR, HIPAA, and industry standards," *Int. Sci. J. Eng. Manag.*, 2022. DOI: 10.55041/isjem00141

[53] H. Peddireddy, "A privacy-first governance architecture for compliant generative AI data pipelines: Regulatory guardrails and data protection frameworks for enterprise AI," *Int. J. Comput. Technol. Electron. Commun. Eng.*, 2024.

[54] B. S. Adelusi, A. C. Uzoka, Y. G. Hassan, and F. U. Ojika, "Reviewing data governance strategies for privacy and compliance in AI-powered business analytics ecosystems," *Glob. Int. Sci. Res. Rev. J.*, 2024.

[55] European Parliament and Council, "Regulation (EU) 2016/679 of the European Parliament and of the Council of 27 April 2016 on the protection of natural persons with regard to the processing of personal data (General Data Protection Regulation)," *Off. J. Eur. Union*, L 119, pp. 1–88, 2016.

[56] European Commission, "Proposal for a Regulation of the European Parliament and of the Council Laying Down Harmonised Rules on Artificial Intelligence (Artificial Intelligence Act)," COM(2021) 206 final, Brussels, 2021.

[57] S. Devarapalli et al., "Cloud-native LLMOps meets DataOps: A unified framework for high-volume analytical systems," in *Proc. IEEE Int. Conf. Cloud Technol. Data Comput. (ICCTDC)*, 2025. DOI: 10.1109/icctdc64446.2025.11158069

[58] S. Mishra, "Trustworthy agentic AI pipelines: Human-in-the-loop oversight architectures for secure enterprise deployment," *ResearchGate Preprint*, 2024.

[59] P. B. Alla, "Human-in-the-loop intelligent automation: Enhancing workflow adaptability through active learning and AI-driven feedback loops," *ResearchGate Preprint*, 2024.

[60] Z. Tiantian, S. Shaowei, X. Mingzhi et al., "Agent-in-the-loop: A data flywheel for continuous improvement in LLM-based customer support," *arXiv preprint arXiv:2510.06674*, 2025. DOI: 10.48550/arXiv.2510.06674

[61] C. L. V. Lopes, J. M. Pitta, F. Belém, and G. Alves, "Engineering AI agents for clinical workflows: A case study in architecture, MLOps, and governance," in *Proc. ACM Conf.*, 2026. DOI: 10.1145/3793653.3793774

[62] R. Dholariya, "Human-in-the-loop AI for cloud data engineering: The Collaborative Intelligence Architecture (CIRA) for regulated industries," *J. Inf. Syst. Eng. Manag.*, vol. 11, no. 1, 2026. DOI: 10.52783/jisem.v11i1s.14188

[63] U. Gadiraju et al., "DataOps meets LLMOps: Automating cloud-based AI workflows from data ingestion to prompt optimization," in *Proc. IEEE Int. Conf. Data Intell. Cogn. Inform. (ICDICI)*, 2025. DOI: 10.1109/icdici66477.2025.11135202

[64] "Generative AI in enterprise data engineering: Integrating Copilot for ETL automation," Zenodo, 2022. DOI: 10.5281/zenodo.17090853

[65] S. Kanagarla, "Data engineering with generative AI: Automating pipelines and transformations using LLMs like GPT," *SSRN Electron. J.*, 2025. DOI: 10.2139/ssrn.5107348

[66] "Next-gen ETL: Integrating large language models with data orchestration tools," Zenodo, 2025. DOI: 10.5281/zenodo.17112926

[67] S. Modalavalasa, "Large language models for intelligent data engineering: Automating schema design, lineage, and quality control," 2024.

[68] "Prompt engineering: A framework for automated SQL generation, schema validation, and anomaly detection using large language models," *Int. J. Artif. Intell. Data Res.*, vol. 16, no. 2, 2025. DOI: 10.71097/ijaidr.v16.i2.1663

[69] A. Zou et al., "What should data science education do with large language models?" *arXiv preprint arXiv:2307.02792*, 2023. DOI: 10.48550/arXiv.2307.02792

[70] H. Takallou et al., "LLMs for end-to-end machine learning: A comprehensive review," 2024.

[71] P. Manjunath and T. Pruefer, "A quantum inspired variational kernel and explainable AI framework for cross region solar and wind energy forecasting," *arXiv preprint arXiv:2605.09032*, 2026. DOI: 10.48550/arXiv.2605.09032

[72] B. Ozer, "Systematic technical survey on LLMOps: Lifecycle, tools, challenges, and emerging practices," 2024.

[73] S. Narayanan, "A generative AI framework for data pipeline optimization and analytical performance enhancement," *Int. J. Artif. Intell. Data Sci. Mach. Learn.*, vol. 6, no. 4, 2024. DOI: 10.63282/3050-9262.ijaidsml-v6i4p116

[74] A. Siddiqi et al., "Saga++: A scalable framework for optimizing data cleaning pipelines for machine learning applications," *ACM Trans. Database Syst.*, 2025. DOI: 10.1145/3771766