International Journal of Computer Information Systems and Industrial Management Applications
ISSN 2150-7988 Volume 18(3s) 2026 pp. 526-539
https://doi.org/10.70917/ijcisim-2026-2377
Article

ADAPTIVE FEATURE ENGINEERING PIPELINES FOR ENTERPRISE AI SYSTEMS USING REAL-TIME LAKEHOUSE DATA PROCESSING AND DATAOPS AUTOMATION

Pankaj Sahani

Purdue University, West Lafayette IN, sahani1987@gmail.com

Abstract: Various transactional applications, IoT nodes, customer engagement, and operational platforms generate large volumes of heterogeneous data that run enterprise AI systems. But the traditional feature engineering methods rely on pre-defined pipelines, manual transformations, and periodic batch processing, making it challenging for AI models to adapt to new data patterns as they emerge. The paper outlines an adaptive feature-engineering framework that leverages the real-time data-processing capabilities of a lakehouse and DataOps automation to scale and continually evolve enterprise AI systems. The proposed model, which integrates all streaming and batch data into a single lakehouse, provides automatic feature extraction, transformation, validation, and deployment throughout the entire AI lifecycle. This approach integrates DataOps practices for automated data quality monitoring, pipeline orchestration, versioning, and drift detection to enhance reliability and streamline operations. Besides, feature pipelines come with adaptive capabilities that enable them to adjust to data changes and evolving business needs while preserving governance and traceability. The objective of the framework is to reduce feature-engineering latency, improve model performance, and enhance scalability in enterprise settings. The study offers a conceptual blueprint and a pathway to implement real-time, data-driven AI solutions with smart, automated feature management tools for organizations aspiring to operationalize artificial intelligence in real time.

Keywords: Adaptive Feature Engineering, Enterprise Artificial Intelligence, Lakehouse Architecture, DataOps Automation, Real-Time Data Processing.

1. INTRODUCTION

Digital transformation projects have grown exponentially and have redefined data collection, management, and utilization, including how enterprises might use data for business intelligence and/or Artificial Intelligence (AI)-driven decision-making. Regardless of industry, organizations today generate an enormous amount of information from transaction systems, customer interactions, mobile applications, sensors, Internet of Things (IoT) devices, social networks, and cloud-based services [1]. In an era of rising data, the ability to turn data into meaning has become a driver of organizational competitiveness and innovation. AI plays a major role in this transformation, employing AI-powered systems for predictive analytics, automation, recommendation systems, anomaly detection, and intelligent decision support [2]. But complex machine learning algorithms are not the only factor affecting AI system performance; the quality, relevance, and representativeness of the data features used throughout the system's development and deployment are also of great importance. In addition to the text's context, one of the most significant factors affecting the accuracy and reliability of a model is the feature engineering of its attributes, in which attributes are extracted, transformed, selected from the data, and enriched with relevant information [3]. A good feature helps the machine learning model detect meaningful patterns, relationships, and trends that are not directly apparent in the raw original data sets. Much of the feature engineering before automation has been done manually, relying on domain knowledge, handcrafted transformations, and data pipelines that move data through a process over time, typically on a batch basis, using Extract-Transform-Load (ETL) [4]. While these methods have been successful in past analytics applications, they have not been effective in enterprise data environments for high-volume, high-velocity, and high-variance data. Recent work on cloud computing, distributed processing systems, and big data technologies has enabled automated, scalable feature engineering techniques that support large-scale applications of artificial intelligence [5]. Meanwhile, much of this information is arriving in various data formats from a range of sources, such as sensors or satellites, and, with a proliferation of data lake or lakehouse designs, data is now being stored, processed, and analyzed by multiple teams across data centers, encompassing a diverse array of departments [6]. Apart from these technological developments, dataOps is emerging as a new paradigm for running data engineering workflows, incorporating automation, collaboration, monitoring, and continuous improvement as best practices, thereby creating an efficient and reliable data-driven system [7]. This blend of AI customers and the synergy of adaptive feature engineering, lakehouse analytics, and DataOps automation will prove attractive for companies seeking to modernize enterprise AI operations and continuously realize intelligence in rapidly evolving enterprise environments.

Many companies still face significant operational challenges when executing feature engineering processes at scale, even with the adoption of AI technologies and modern data platforms [8]. Many machine learning pipelines assume that the feature definition remains fixed and is hard-coded into the models themselves when they are deployed. They are typically unable to handle evolving business needs, fluctuating customer behavior, market shifts, and concept drift, resulting in a decline in model performance over time. Furthermore, many organizations still rely on "siloed" data systems (typically five or more) that are all massive, separate databases, data warehouses, etc., that are designed to handle a single type of data, yet forfeit a key component of the strengths of data: it flows easily, and essential features can easily be created efficiently [4]. The batch-based processing models rely on data generated at a particular moment and do not use it to respond to actual events or changing operational conditions in real time, which can severely undermine an AI system's ability to adjust its response time to events [9]. As streaming data technologies have become increasingly popular, so has the need for low-latency feature generation, although many existing implementations lack a means to continually update and validate features in production. Furthermore, data quality and integrity, changing data schema structures, inconsistent attribute definitions, metadata meaning, and data repeatability across multiple AI projects pose problems for organizations [7]. While the feature store is a key pillar of today's machine learning landscape, it is used across a variety of deployments, with feature reuse occurring without consideration of adaptations throughout the feature life cycle. Similarly, DataOps frameworks have proven effective at increasing pipeline observability and automation, but have been poor at adopting intelligent feature engineering processes so far. Similarly, the idea of coupling Intelligent Feature Engineering (IFE) processes with DataOps frameworks is yet to gain traction [10]. While some prior studies studied automatic feature engineering, automatic machine learning (AutoML), stream processing, or the lakehouse architecture independently and in isolation, few studies looked at the application of these technologies together and evaluated both their ability to adapt features as data changes over time while also meeting varying business needs and properties. Therefore, it becomes difficult for firms to preserve these two to four characteristics while maintaining one or more of the following features as part of their continuing strategy: turnover, vitality, well-governed, and effective to run. Therefore, some companies are struggling to keep their features fresh, scalable, healthy, and well-governed in their operations. This is where a fully-featured architecture based on adaptive feature engineering, real-time data processing, and DataOps automation comes into play for next-gen enterprise AI systems.

These challenges lead to a new trend in the industry: Lakehouse architectures and DataOps practices that may provide a starting point for transitioning to smarter ways of feature engineering. Lakehouse platforms combine the best features of data warehouses and data lakes, such as scalable storage, transactional consistency, schema support, analytical and machine learning workloads, and so on, all under a single roof [6]. Meanwhile, DataOps introduces automation that facilitates data integration, quality checks, monitoring, testing, deployment, and governance throughout the data lifecycle [7]. Together, these technologies can develop adaptive feature engineering pipelines that automatically adapt to changes in the data, evolve feature representations as new information becomes available, validate feature quality, and update feature stores in real time. In enterprise environments, where AI models need to consistently adapt to varying factors such as changing operational requirements, customer tastes, or market dynamics, these capabilities become more crucial than ever. This paper, therefore, proposes an adaptive feature-engineering approach that dynamically processes data in the lakehouse and optimizes the DataOps pipeline using AI to enhance the scalability, reliability, and responsiveness of enterprise AI systems. The overall objective of the framework is to seamlessly integrate feature generation, transformation, validation, monitoring, governance, and deployment on a common platform and automate the process. The proposed approach combines real-time analytics with an intelligent feature lifecycle to minimize feature-engineering latency, enhance feature quality, enable continuous model improvement, and boost enterprise AI decision-making capabilities. Furthermore, this study demonstrates the promise of adaptive feature pipelines to bring together data, ML, and enterprise-level AI operationalization in one place, helping to deepen understanding of how these technologies can best be integrated. Moreover, this is one of the first studies in the literature to provide a practical example of how feature pipelines can be applied to integrate data engineering and machine learning engineering for enterprise-level AI deployment.

Scope of the Paper

1. Develop the essence of feature engineering as a concept and problem in the enterprise AI landscape in today's world.
2. This will address the value of real-time lakehouse architectures for uniform, low-latency, and scalable feature processing.
3. To look at how DataOps automation can be used to optimize feature lifecycle through continuous monitoring, validation, orchestration, and governance.
4. To propose an adaptive feature engineering process using streaming, lakehouse, feature stores, and DataOps.
5. To evaluate the proposed framework for feature freshness for feasibility, performance of the AI models, operational efficiency, scalability, and enterprise AI readiness.

2. RELATED WORK

The skills required in enterprise AI solutions were shifting toward identifying ways to boost machine learning efficiency through feature engineering, scalable data architectures, and automated data operations. In early works [8], the feature engineering technique, in which features were manually extracted, transformed, and selected in the structured data space, was the predominant approach. These methods worked well in this example to predict how well their model would perform, but required substantial talent to implement and were hard to scale to large collections. Later studies implemented machine learning methods, such as automatic feature engineering and feature synthesis, to minimize human intervention in model development and accelerate the process [9]. However, as the big data ecosystem emerged, and big data needed to be processed in real-time, the researchers made an effort to research cloud-native infrastructures, different distributed computing systems that should accommodate them, etc. [10]. A new wave of data architecture, data lakes and lakehouse data architectures, emerged again and stimulated research to find a unified data management platform for analytical workloads and machine learning [11]. The importance of feature stores, metadata management, and feature reuse mechanisms for better consistency of machine learning pipelines has been noted in recent studies [12]. Concurrently, another promising DataOps approach aims to enhance data collaboration and automation capabilities and to monitor and govern the entire data life cycle [13]. Most recent work has focused on live feature pipelines, streaming analytics, and adaptive systems that dynamically learn from variations in concepts and data structures [14]. But after these changes, most accessible books and resources focus on feasible space components and architectures, or on DataOps automation within the lake. Very little research has been done on adapting feature engineering to a "real-time processing application" (i.e., as part of a single application) with DataOps-style orchestration [15] across the lakehouse as a whole. Much work is underway to create architectures that are scalable, controlled, and efficient for enterprise systems, which are dynamic and continuously add features that need to be invalidated, monitored, and managed.

Table 1. Comparative Analysis of Existing Related Studies

Ref.	Research Focus	Methodology/Technology	Key Contribution	Limitations
[8]	Traditional Feature Engineering for Machine Learning	Manual feature extraction and selection techniques	Demonstrated importance of engineered features in improving model accuracy	Highly dependent on domain expertise and difficult to scale

[9]	Automated Feature Engineering	Feature synthesis and automated transformation pipelines	Reduced manual effort in feature generation	Limited adaptability to real-time data streams
[10]	Big Data Processing for AI Systems	Distributed computing and cloud analytics frameworks	Improved scalability for large-scale machine learning workloads	Primarily focused on data processing rather than feature lifecycle management
[11]	Lakehouse-Based Analytics Architecture	Unified storage and analytics platform	Combined advantages of data lakes and data warehouses	Limited support for adaptive feature generation mechanisms
[12]	Enterprise Feature Stores	Centralized feature repositories and metadata management	Enhanced feature consistency and reuse across models	Feature updates often remained static and batch-oriented
[13]	DataOps for Data Pipeline Automation	Continuous integration, monitoring, and orchestration	Improved operational efficiency and data quality management	Limited integration with intelligent feature engineering processes
[14]	Real-Time Feature Pipelines	Stream processing and low-latency feature generation	Enabled near real-time machine learning applications	Governance and feature drift management challenges remained
[15]	Adaptive Machine Learning Systems	Dynamic model adaptation and drift-aware learning	Improved responsiveness to changing data environments	Focused mainly on model adaptation rather than feature adaptation
[11,14]	Streaming Lakehouse Architectures	Real-time analytics integrated with lakehouse storage	Supported unified processing of batch and streaming data	Limited DataOps-driven automation capabilities
[12,13,15]	FeatureOps and Operational AI Frameworks	Feature lifecycle management with automation practices	Improved reproducibility and deployment efficiency	Lack of comprehensive integration across feature engineering, lakehouse processing, and DataOps automation

3. PROPOSED ADAPTIVE FEATURE ENGINEERING FRAMEWORK

The future of enterprise AI systems lies in meeting the ever-changing needs of real-time analytics and intelligent decision-making, demanding systems that are not only expanding in size but also in complexity. As the business world embraces real-time analytics and intelligent decision-making, enterprise AI systems must not only keep growing in size but also in complexity. Traditional methods of feature engineering are often constrained by static workflows, periodic batch updates, and manual effort to solve problems, which makes them less suited to the evolving nature of business settings and patterns. To mitigate these challenges, this paper introduces an Adaptive Feature Engineering Framework that leverages real-time lakehouse data processing and DataOps automation to generate a never-ending stream of features. The framework aims not only to automate the feature collection, transformation, validation, and monitoring process but also to maintain consistency between different environments and spaces in machine learning production and development. The Unify Lakehouse Storage component allows for the seamless integration of streaming and batch data sources into a common architecture, further enhancing the framework's capabilities. To further strengthen the framework capabilities, the Unify Lakehouse Storage component streamlines data processing and enables the seamless integration of streaming and batch data sources. The incoming data streams are constantly monitored, and the algorithms detect changes in data distributions, feature relevance, and operational demands, thereby automatically updating features and mitigating data model degradation caused by feature drift. The DataOps principles have been integrated to enable efficient delivery of feature pipelines, ensuring reliability and reproducibility throughout the AI lifecycle, including continuous integration, testing, orchestration, monitoring, and governance. Additionally, the framework supports centralized feature management via feature stores, which enable feature reuse, feature metadata tracking, version control, and consistent training and inference. These are integrated into a scalable solution that can power a variety of enterprise solutions from the fraud floor to customer intelligence and analytics, predictive maintenance to healthcare intelligence, and, going forward, to supply chain optimization. The proposed framework takes a more holistic approach to feature engineering, focusing on adapting, automating, and managing the feature lifecycle. Organizations can thus achieve greater feature freshness, faster model deployments, better predictive performance, and build a scalable infrastructure for enterprise machine-learning operations in today's rapidly changing, data-heavy landscape.

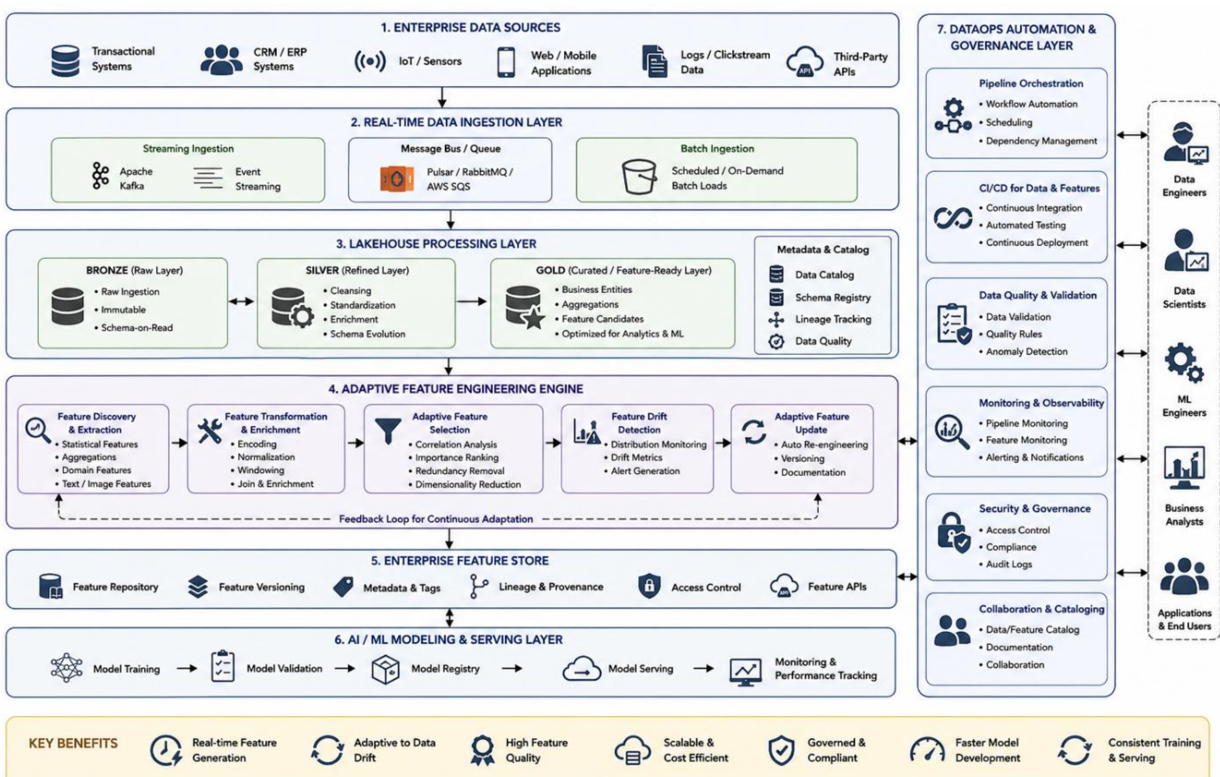

Figure 1. Proposed Adaptive Feature Engineering Framework for Enterprise AI Systems

3.1 Real-Time Data Ingestion Layer

The Real-Time Data Ingestion Layer is the first layer in the proposed framework and is responsible for gathering data from various enterprise sources. Information flows in modern organizations from operations involving customers, inventory, order-taking, enterprise resource planning, customer relationship management applications, Internet of Things sensors, Web applications, mobile services, and external Application Programming Interfaces (APIs). Many data sources provide data at different rates and formats, making it difficult for traditional data ingestion methods to handle. These challenges are overcome with a hybrid ingestion architecture that can ingest both streaming and batch data. Streaming ingestion captures events as they occur, enabling AI systems to respond faster to operational changes; batch ingestion is suitable for periodic updates to larger and historical datasets. The ingestion layer features automated schema discovery, data validation, and metadata-capture capabilities to ensure consistency as data is ingested into the lakehouse environment. Event-driven processing mechanisms also help to identify important business events that can initiate an event; these events could represent feature and/or analytical workflow updates. The ingestion layer enables continuous ingestion of the data that powers the enterprise, serves as the foundation for generating features on the fly, and provides adaptive analytics capabilities. Furthermore, the architecture is scalable, and the web service can be scaled to a larger data plantation by using distributed processing techniques that increase data volume without affecting application performance. Monitoring and logging are built-in features to see ingestion performance, latency, and reliability. This translates into high-quality, complete, and timely data flows to enterprise AI systems, enabling adaptive feature engineering and intelligent decision-making.

3.2 Lakehouse Processing Layer

The core part of the envisioned framework is the Lakehouse Processing Layer. It combines the flexibility of data lakes with the governance, reliability, and transactional capabilities of data warehouses. A single architecture that removes the need for separate analytical repositories while enabling processing of multiple workloads, such as business intelligence, machine learning, and real-time analytics. The data entering the lakehouse is structured through several stages of refinement. The Bronze Layer will preserve source information and retain the raw data in the original format of incoming events to ensure traceability. The Silver Layer is responsible for data cleansing, standardization, integration, and enrichment for data quality and consistency. Last, and most importantly, the Gold Layer provides business-ready, feature-ready datasets for analytical and machine learning tasks. The lakehouse architecture provides both batch and streaming capabilities, with data processes running in the same space, enabling organizations to analyze real-time and historical information from a shared lake. Its advanced features, like schema evolution, version control, transactional consistency, and metadata management, enhance data reliability and governance. A unified structure can drastically cut expenses by reducing data duplication and enabling easy access by other enterprise departments. Additionally, the lake house can serve as a building block for adaptive feature engineering, given the availability of lake-produced datasets that are refined and reflect current operational conditions. The Lakehouse Processing Layer empowers businesses to efficiently generate features while maintaining governance and quality standards for enterprise AI deployments, all through centralized data management and scalable processing capabilities.

3.3 Adaptive Feature Engineering Engine

The key innovation of the proposed framework is the Adaptive Feature Engineering Engine. The advantage of the proposed engine is that it is self-extensible to meet business requirements and data properties; it differs from traditional feature engineering systems that use a fixed set of transformations or a library of known ones. The engine extracts, transforms, aggregates, enriches, and selects features from the real-time and historical data flowing from the lakehouse environment. Machine learning applications also receive statistical features, temporal features, behavioral indicators, contextual attributes, and domain-specific variables to automatically support their operations. Continuous monitoring mechanisms assess a feature's relevance, stability, and predictive contribution over time. If there are frequent changes in data distribution or feature importance, the engine will make adaptive adjustments to preserve feature effectiveness. Drift detection algorithms are used to alert to degradation in feature quality, and, when needed, reengineering processes are employed. Automated selection mechanisms select the most informative features whilst discarding redundant or less valuable attributes. Moreover, the metadata generated during feature creation is recorded to promote transparency, reproducibility, and governance. The engine directly interfaces with feature lifecycles, reducing manual processing and enabling enterprise AI methods to adapt to evolving operating contexts. This is a powerful capability that can help boost accuracy, increase model robustness, and ensure the sustainability of machine learning deployments in various enterprise use cases.

3.4 Enterprise Feature Store

The Enterprise Feature Store serves as a single source of truth for engineered features, managing them across the AI lifecycle. It's one of the major issues in enterprise machine learning systems: keeping data consistent during training and inference and enabling feature reuse across different projects. The idea for the feature store is inspired by these challenges; it's meant to be a single place to store, store catalog, version, and share engineered features. For each feature, diagnostics include data on its origin, transformations, update frequency, and various quality and usage metrics. Version control enables one to go back through history to reproduce feature definitions and provide audits, ensuring access to historical feature definitions. The feature store is designed to be accessible both online and offline. Model training and experimentation use offline features, whereas online features provide low-latency access for real-time inference applications. Centralized governance capabilities help organizations track feature usage, ensure compliance, and maintain data quality. Additionally, feature sharing simplifies the work of multiple business units and helps to standardize the enterprise's operations. These lineage features enable stakeholders to see the entire story of each feature as it goes from source data to deployment. With these features, the Enterprise Feature Store enables operational efficiency, fosters collaboration among data teams, and facilitates organization-wide deployment of AI solutions.

3.5 DataOps Automation and Governance Layer

The DataOps Automation and Governance Layer offers operational intelligence, reliability, and governance for the entire adaptive feature engineering framework. The guiding principles of DataOps focus on delivering the benefits of automation, collaboration, continuous learning, and quality control across all data-centric workflows. This layer orchestrates feature engineering activities, including automated pipeline orchestration, monitoring, testing, deployment, and validation within the proposed architecture. Feature pipelines can be updated efficiently and often in the background, thanks to continuous integration and continuous deployment processes. Automated quality assurance checks the completeness, consistency, accuracy, and timeliness of data prior to the availability of features for model consumption. Monitoring systems continuously track the health of features, pipeline performance, and latency, as well as issues such as operational anomalies, enabling early detection and resolution before problems occur. Governance mechanisms are established in the enterprise environment to facilitate regulatory compliance, auditability, access control, and policy enforcement. Lineage and metadata capabilities further enhance transparency and accountability throughout the entire AI lifecycle. What's more, it provides automated alerting and incident management, which decreases the dangers of feature degradation or system failures. The adoption of DataOps principles, combined with adaptive feature engineering workflows, can improve the reliability, scalability, and efficiency of data projects while upholding strong governance and compliance standards. This layer effectively turns feature engineering into an enterprise capability that is continuously managed and automated, facilitating sustainable AI operations.

4. EXPERIMENTAL DESIGN

The study's design is designed to comprehensively evaluate the proposed Adaptive Feature Engineering Framework in modern enterprise AI applications. One of the problems facing modern feature engineering methodologies is the data's velocity, volume, variety, and ever-changing nature. They typically build feature engineering pipelines following the design philosophy of batch design - that feature transformations are defined, and that periodic data processing is leveraged. While these solutions were previously used to support predictive analytics applications, they tend to offer few advantages in terms of speeding up feature availability, adapting to new trends, reducing maintenance workload, and dealing with concept drift. Therefore, the need for an adaptive feature engineering mechanism that is sufficiently adaptable to varying data patterns, consistent and controlled, yet efficient to run, is growing. The proposed framework has been tested in a multi-dimensional experimental environment that leverages heterogeneous datasets gathered from both transactional databases, as well as from IoT sensors, CRMS (customer relationship management software), ERP (enterprise resource planning software), from web applications, clickstream logs, and external data services, in realistic enterprise environments. These datasets include enterprise operations and business scenarios, such as structured, semi-structured, and event-driven data. Our experimental setup includes a few workloads that form a realistic enterprise data flow with a changing schema, dynamic data inflows, varying sizes, and varying operation patterns. There are three Architectural scenarios to be compared. The classic look of a batch-engineered feature+feature engineering looks like Scenario A: it often needs a feature update and requires manual feature engineering. No use of a batch and streaming data integrator, modern lakehouse framework, and statically defined features, as in Scenario B. For scenario C, the data difference is typically addressed alongside the feature meta-portfolio (as proposed in this paper), feature drift monitoring, metadata-driven governance, continuous feature extraction, automatic feature recalibration, and orchestration in DataOps. The key is that the same datasets and model are used across all situations, and multiple machine learning tasks are performed (including predicting customer churn, detecting anomalies and fraud, predictive maintenance, and forecasting demand). We'll use a number of metrics across data engineering, machine learning performance, feature lifecycle, operational efficiency, and governance. The experimental design integrates analysis, predictive operations, and infrastructure to evaluate the impact of adaptive feature engineering on enterprise AI system performance, feature sustainability, deployment agility, and system longevity and reliability.

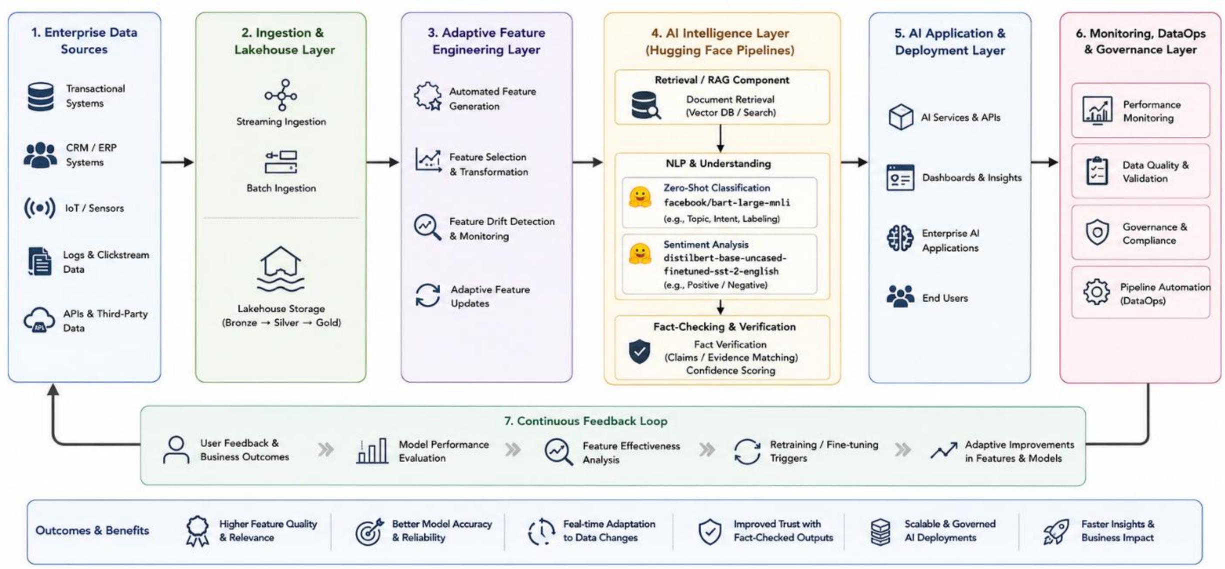

Figure 2. Experimental Workflow for Adaptive Feature Engineering Evaluation

In addition to the model accuracy assessment approach, it also covers in-depth observation of the model's behavior during the feature lifecycle, operability and automation, enterprise obstacle management characteristics, and scalability. Experiments are conducted across multiple execution cycles to capture the dynamics of enterprise environments under varying operating conditions and load intensities. As time goes on, during the evaluation period, a lot of data can be gathered, which corresponds to several interactions that the customer had with the system, a number of transactions processed in the system, reading on other types of IoT sensors, etc., as well as events and operations of systems and business processes. There are several types of deliberately injected data drift, such as concept drift, feature drift, seasonal and behavioral changes/schema changes, and change of transaction frequency distributions. Among the scenarios are various business scenarios, requiring the AI to be flexible in responding to different business situations within an organization. Adaptive feature engineering continuously monitors data streams using statistical analysis methods such as statistical profiling, distribution analysis, feature importance monitoring, correlation monitoring, and drift detection. An important deviation in any feature parameter from the baseline features initiates feature recalibration and a transformation update, and synchronizes metadata. A comparative experiment is conducted to assess how quickly the reaction time to changes in the data condition is, and how this affects adaptive updates that improve predictive capabilities. The DataOps layer also automates the experiment lifecycle, from experiments' data and schema validation to pipeline quality and monitoring, from orchestration to lineage tracking, and up to experiment deployment. To test the system's scalability, data volume, number of concurrent users, data storage, and processing complexity were incrementally increased to assess the system's elasticity and ability to process data effectively. Other operational metrics are pipeline runtime, deployment rates, incident recovery time, monitoring coverage, governance compliance, and feature reuse rates. The following features are measured at the performance level: feature freshness, feature statefulness index, feature redundancy ratio, feature availability, and drift response latency. Furthermore, the experimental design assesses reproducibility, feature version control, and metadata completion. These tests collectively provide a comprehensive suite of tools to assess the impact and application of adaptive feature engineering in enterprise AI, ensure consistency of analysis across the board, improve system resource efficiency, and enhance overall sustainability. Their experimental results inform their application of a proposed framework for a large organization operating in dynamic data ecosystems and with numerous AI deployment needs.

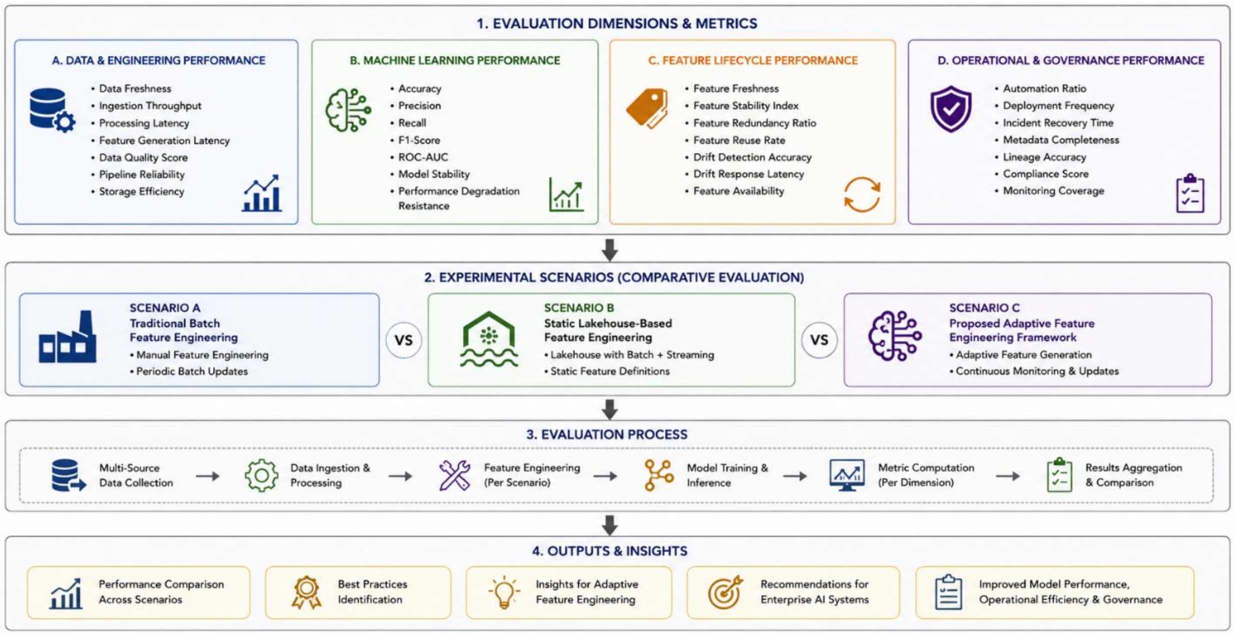

Figure 3. Comprehensive Performance Evaluation Framework

5. RESULTS AND DISCUSSION

The value of leveraging real-time processing and adaptive feature engineering in the lake, all the way through to data delivery into the data pipeline integrated with the “deliver” DataOps automation end-to-end, has been proven in the enterprise AI ecosystem. Both works that sought to reduce the number of stateful feature pipelines, to craft features manually, and to give machine learning systems greater flexibility in the context of the data were experiencing networking phenomena. The data distributions also vary over time [16, 17], and so do observations of these distributions, which may enhance feature relevance or representation (this is known as adaptive feature engineering), helping in other scenarios (any mechanism) as well. Others add value by minimizing the time lag between analysis/operation and feature production, as well as by providing more real-time information from analysis/operation. Feature stores and FeatureOps practices aim to increase feature consistency, eliminate feature duplication, provide centralized management of feature stores and feature metadata, and track feature lineage [18]. Moreover, numerous issues in getting streaming data into enterprise data stores, which were common a few years ago, have been dramatically solved, and streaming data can be delivered as well as historical data. Also, easy access not only to the streamed data streams but to past DTO data as well, thereby removing many of the obstacles to streaming data integration with legacy enterprise data stores. DataOps automation enhances these capabilities through continuous validation and monitoring, deployment automation, and governance, all of which foster reliability and reduce operational complexity [19, 20]. Throughout the machine learning lifecycle, these technologies yield a “magic bullet” that improves model sustainability, scalability, and governance. In these cases, organizations tend to be more successful in integrating adaptive feature engineering, enabling them to adapt to a rapidly evolving marketplace with increasing amounts of data and evolving analytical requirements. It has a bias towards making more substantial contributions to current-day-to-day business AI systems and suggests some more appropriate indicators: “freshness of features”, “deployment agility”, “observability in operation”, and “governability of features”. Put together, this approach to developing adaptive feature engineering on top of the lakes and this embrace of the DataOps thinking creates a real-world situation where they need to make decisions at an early stage, where technical debt will likely be in play, and where they won't be able to reap the benefits of the successes of AI in the same way.

Table 2. Results Analysis of Enterprise AI Capabilities

Evaluation Area	Traditional Approaches	Modern Enterprise Approaches	Observed Outcome
Feature Engineering	Manual and periodic updates	Adaptive and automated updates	Improved feature relevance and reduced maintenance effort
Data Processing	Batch-oriented workflows	Real-time and streaming processing	Faster analytical response and improved data freshness
Feature Management	Project-specific feature creation	Centralized feature stores	Higher feature reuse and consistency
Data Infrastructure	Separate warehouse and analytics systems	Unified lakehouse architectures	Better scalability and accessibility
Operational Management	Manual monitoring and deployment	Automated DataOps workflows	Increased reliability and reduced operational overhead
Governance	Fragmented metadata management	Centralized lineage and monitoring	Improved traceability and compliance
AI Lifecycle Support	Static model environments	Continuous adaptation mechanisms	Greater model sustainability

Though the original objective of machine learning environments was to make better predictions by selecting and transforming features, in the enterprise space today, platforms aiming at machine learning environments must support a more scalable environment built on sound feature selection and transformation to be more governable and sustainable throughout a platform's life cycle. But it is not enough to rely on conventional feature engineering methods in general, since the features can be continuously monitored and updated (if needed) with drift-detection capability, and automatically maximized [16] in an adaptive way. Real-time Processing infrastructures reduce delays, and features can be offered on demand, leading to the "Application" where real-time analysis is needed [17]. By sharing feature definitions and shared feature assets across different projects [18], Office automation and feature stores can operate efficiently and be up and running quickly. Likewise, lakehouse architectures offer a single platform that can handle a variety of analytical workloads and make data management easier [19]. DataOps practices also play a role in ensuring reliability and control over enterprise AI environments, including continuous integration, automated validation, deployment orchestration, and observability. Companies with an integrated architecture demonstrate the potential to address complexity challenges, roll out company-wide AI solutions, and provide long-term application performance stability, whereas those without seem to diverge from the results of companies with such an architecture. Dynamic feature adaptation, governance of feature assets, and the ability to automate processes will soon be crucial capabilities for enterprise AI systems, enabling AI applications to be sustainable and scalable.

Table 3. Comparative Evaluation and Framework Alignment

Dimension	Existing Enterprise Approaches	Key Challenge	Proposed Framework Capability
Feature Generation	Automated but largely static	Feature degradation over time	Continuous adaptive feature generation
Feature Freshness	Periodic updates	Delayed response to data changes	Real-time feature updates

Feature Governance	Centralized feature repositories	Limited adaptive management	Dynamic feature lifecycle governance
Data Processing	Streaming and batch integration	Fragmented feature intelligence	Feature-aware lakehouse processing
Operational Automation	DataOps and MLOps practices	Weak feature engineering integration	End-to-end orchestration
Drift Management	Basic monitoring mechanisms	Limited feature recalibration	Automated drift-aware adaptation
Scalability	Distributed processing architectures	Growing operational complexity	Unified adaptive ecosystem
Enterprise AI Sustainability	Continuous deployment pipelines	Long-term feature maintenance challenges	Lifecycle-driven feature management
Compliance and Traceability	Metadata and lineage frameworks	Governance fragmentation	Centralized governance and monitoring
Overall Enterprise Readiness	Modernized AI infrastructure	Lack of holistic integration	Fully integrated adaptive architecture

6. DISCUSSION

The results of this study indicate that adaptive feature engineering is a crucial capability that enterprise AI systems must possess in contexts with ever-changing data and requirements. The traditional approach to feature engineering has been developed mainly for data with relative stability and/or for specific analytical data-processing tasks conducted periodically over long periods, with consistent feature definitions. Businesses collect data rapidly from a multitude of sources (including customers, operations, IoT devices, digital platforms, and external interactions and services), and managing this data is essential to business operations. In these cases, the basic set of feature pipelines supplied in static app implementations will not be flexible enough to quickly accommodate changing data delivery requirements, evolving behaviors, and changing business. The results show that these drawbacks can be taken care of efficiently with a feature engineering approach of adaptation, and once adaptive features are monitored, they can be automatically recalibrated, lifetime, and then monitored and adapted (as the test time progresses). Furthermore, even though all the data is processed in a single location, real-time lakehouse architectures are better suited to efficiently leverage adaptive feature engineering. Lakehouse platforms provide more scalable, convenient navigation, storage, and processing capabilities than data architectures that rely on multiple isolated storage and processing platforms. With feature stores, features such as feature consistency, metadata, and the ability to reuse and reproduce them can be incorporated into machine learning workflows. The developments are the ascent in all those functional reductions in functional inputs, and conductive advances in the construction of sound AI. The findings also revealed that feature engineering is not limited to the data preparation phase but can continue to improve performance over time, enabling enterprise AI to become sustainable. As AI becomes more widely used to aid in data-driven decisions within organizations, adaptive feature engineering's requirement for its accuracy, for little or no model degradation-perception, and for intelligence generation on the fly will grow. Hence, the proposed model is an appropriate extension of conventional feature engineering techniques towards more powerful, scalable enterprise-tech-type AI systems that are increasingly being deployed.

However, from an operations perspective, the study reiterates that automation plays a role in the DataOps equation, translating the concept of adaptive feature engineering from theory to practice as an enterprise service. Among the most frequently occurring enterprise artificial intelligence issues are data pipelines, ingestion, validation, processing, monitoring, and real-time data delivery to various analytical systems. These activities can be time-consuming, may have inconsistencies, can cause governance problems, and can be more time-intensive to manage. DataOps tackles these attributes by integrating automated orchestration, continuous validation, monitoring, observability, and governance processes to deliver reliability and efficiency. This monologue will also address the need for all parts of the Enterprise AI system to remain in sync across components: data engineering, feature engineering, machine learning ops, and governance. Previously, the relationship among the above components has been studied independently; however, these components need to be better coordinated, particularly in the data environments of enterprises with more elaborate structures. This includes all newly launched feature-generation capabilities built with adaptive, built-in support for processing on Lakehouse, along with new central feature management and automated controls for operations, all packaged into one. Companies can gain advantages, including increased reliability in maintaining current features, reduced deployment delays, enhanced compliance tracking, and smoother lifecycle tracking. Further, the framework can serve as a foundation for new enterprise AI requirements, such as self-learning systems, autonomous analytics, real-time decision support, and intelligent process automation. While these are benefits, there remain important and relevant questions about the near future that are worth considering, such as cross-platform interoperability, effective use of computational resource optimization, and a more in-depth study of governance control policies for large-scale implementation. The outcome, though, suggests that companies that adopt engineering best-practice principles and automation based on DataOps will be more likely to achieve scalable and sustainable AI adoption, and that business continuity in AI operations is likely to be achieved by those that embrace adaptive “lakes on lakes” in the rapidly evolving digital world.

7. CHALLENGES AND LIMITATIONS

While the Adaptive Feature Engineering Framework has great potential for enterprise AI environments, it also presents technical, operational, and organizational challenges. Enterprise data ecosystems are inherently complex, spanning diverse data sources, multiple processing platforms, distributed storage systems, and evolving business needs. Incorporating adaptive feature engineering with real-time lakehouse architectures will add several layers of complexity that can influence how viable a deployment is to operational efficiency. Feature monitoring/tracking, generation, validation/correction, and updating are highly compute-intensive processes that require substantial resources, particularly when handling a large number of data flows in a stream and frequent transaction processing in organizations. However, adaptable systems need high-quality, reliable, and consistent data flows; they may suffer from poor data quality, data inconsistencies, and failures in data infrastructure and metadata. While automated governance mechanisms prove handy for monitoring and compliance, it is difficult to maintain transparency and interpretability in the dynamics of emerging features. A common difficulty is deploying multiple technologies, such as lakehouse platforms, feature stores, stream processing engines, and DataOps orchestration tools, which can lead to interoperability challenges and complexity. Furthermore, if feature descriptions are constantly changing and organizations are dealing with regulatory and audit requests, they may struggle to establish a generic governance process that can accommodate those changes. The success of adaptive feature engineering also highly depends on the accuracy of the drift-detection routine and on how drifts are addressed (adapted) through changes to the features, which may miss complex feature-space drift. Despite its benefits, a few challenges must be addressed with care to help organizations overcome them and use AFLM effectively without compromising enterprise AI solutions. Technical, governance, scalability, and operational challenges must be carefully managed for organizations to successfully implement and integrate AFLM into their existing systems to enable enterprise AI.

7.1 Scalability and Computational Complexity

The major difficulty in adaptive feature engineering is the computational resources needed to generate, evaluate, and adapt features in real time. Unlike conventional feature engineering methods, which run periodically, adaptive systems run continuously and need to ingest large data sets of streaming and historical data and process them simultaneously. With enterprise data volume growing, there can be increasing pressure on storage, processing, networking, and memory resources for feature-generation pipelines. Other activities, such as continuous drift detection, feature monitoring, metadata synchronization, and automatic recalibration, also add to computational complexity. When rolling out features in a large-scale enterprise, where millions of transactions or sensor events may occur each day, achieving low-latency feature generation and high system performance can be challenging. Also, distributed processing systems and complex orchestration mechanisms may be needed to implement distributed workload balancing and resource assignment in adaptive systems. These can make infrastructure more expensive and place additional operational pressures. Given this limited supply of computing power, it's possible that adaptive feature engineering may be less easy for organizations to implement. As a result, achieving feature flexibility while addressing computational silos has remained a significant challenge for enterprise-scale deployment.

7.2 Data Quality and Feature Reliability

The effectiveness of adaptive feature engineering relies heavily on high-quality, reliable data sources. In most enterprise scenarios, records are often incomplete, inconsistent, poorly formatted, missing values, or frequently changing, which can obfuscate feature generation workflows. Poor-quality data can introduce inaccuracies from collection to analysis and negatively affect all aspects of the machine learning lifecycle, from model creation to deployment. Adaptive systems constantly extract features from incoming data streams, so if the data is of poor quality, inaccuracies propagate throughout the entire machine learning lifecycle, from model training to deployment. Furthermore, dynamically generated features can be unstable when the properties of the data sources vary considerably over time. Some types of degradation can be detected by feature drift detection mechanisms, but there may be situations where they cannot distinguish between a true business change and brief abnormal conditions. Inadequate data validation processes and inconsistent metadata management can further call the reliability of features into question. Arguably, real-time environments are more susceptible than others because mistakes during data transfer can quickly affect downstream analytics and decision-making systems. So, it is vital to have strong data quality management, validation, and monitoring processes in place to support successful adaptive feature engineering. While consistently producing high-quality feature generation across a variety of enterprise data sources remains a significant challenge, there has been substantial work on automating DataOps.

7.3 Governance, Compliance, and Explainability

One of the primary issues for organizations with adaptive feature engineering systems is governance and regulatory compliance. In highly regulated contexts, where transparency, accountability, and an audit trail are essential, Enterprise AI systems are more often deployed in critical applications that rely on these qualities. In traditional feature engineering workflows, feature definitions are static and easily documented and reviewed. Contrast these with adaptive feature engineering, which continually updates feature representations to keep pace with evolving data; tracking feature evolution and maintaining comprehensive documentation becomes more complicated. For regulatory purposes, there is a frequent need for clear explanations about the impact of particular features on the decision-making process. Because features may change over time, feature adaptation can make explainability more challenging. In addition, companies need comprehensive lineage logs, access management, metadata databases, and compliance monitors to meet compliance standards. Feature stores and DataOps frameworks can help to strengthen governance capabilities, but it's difficult to achieve consistent governance policies across systems and teams. Measuring up against emerging regulatory standards and maintaining the capability for adaptive feature engineering then become major constraints to overcome when deploying and operating a system.

7.4 Integration and Operational Complexity

The proposed framework can be built using a combination of technology enablers: real-time ingestion, the lakehouse architecture, feature stores, stream processing engines, machine learning pipelines, and DataOps automation tools. Each technology can be of great use in the product, but must be effectively integrated into a single architecture, which entails significant operational complexity. Educational and enterprise organizations are often deployed in environments that include third-party services, cloud implementations, and legacy systems that do not necessarily play well with today's adaptive feature-engineering infrastructure. Interoperability issues can arise from differences in data, data communication, data governance, and deployment approaches. Furthermore, the need to synchronize feature stores, metadata repositories, monitoring systems, and machine learning environments requires powerful orchestration capabilities. There also needs to be experts and practitioners in various areas such as data engineering, machine learning engineering, platform operations, and governance management. The simultaneous need for multiple disciplines may lead to higher implementation costs and longer deployment timelines. In addition, failures in a highly connected environment can be harder to troubleshoot than in a more traditional one. Therefore, the complexity of integration and operational management remains a significant consideration when implementing adaptive feature engineering solutions at enterprise scale.

8. Conclusion

As enterprise data ecosystems continue to expand and the use of Artificial Intelligence (AI) grows, challenges of traditional feature engineering techniques using static pipelines, manual work, and batch processing at fixed intervals have become apparent. These transactional systems, user feedback, and interactions with internet of things (IoT) devices, any digital platform, etc., were all collecting a large amount of information that will evolve over time, and with it comes a growing demand for predicting the data change, with solutions that are scalable, reliable, and governed. To meet the needs of their enterprise AI environments, the study proposes an Adaptive Feature Engineering Framework that manages features across all phases of the lakehouse lifecycle, provides data-centralized feature storage, and automatic DataOps. With a simple shift in the engineering perspective, many of the feature-engineering issues faced in traditional systems can be alleviated by the proposed framework, including hidden architectures, heterodox features, feature drift, the challenge of updating features, and governance-related limitations. The lakehouse architecture offers advantages, including the ability to combine batch and/or streaming data in a single process, making data more accessible and readily available for real-time analytical workloads. Adaptive feature build support also enables feature monitoring, feature recalibration in response to new data distributions and changing business requirements, and feature updates to continue on an ongoing basis. Moreover, the automated validation, orchestration, monitoring, deployment, and governance features of DataOps contribute to operational efficiency, enabling a more reliable and sustainable enterprise AI system. Based on the results and discussions, it can be concluded that adaptive feature engineering can increase feature freshness, model robustness, deployment agility, and governance across the model lifecycle, while also freeing up time from manual effort and/or operational overhead. Overall, the proposed framework serves as a good starting point for tackling challenges and meeting the requirements of future AI-based ecosystems. In summary, the proposed framework is a good starting point for addressing some of the challenges and works towards meeting the needs of future AI ecosystems. Feature discovery from unsupervised data, feature optimization using AI, self-healing data pipelines, and complex learning mechanisms, including drift detection, are other possible avenues for research. Last but not least, custom feature engineering using a lakehouse architecture and dataOps automation are emerging as potential strategies for building scalable, resilient enterprise AI systems that enable real-time, data-driven decision-making.

References

1. Department of Economics. (2024). *United Nations E-Government Survey 2024: Accelerating Digital Transformation for Sustainable Development-With the addendum on Artificial Intelligence*. Stylus Publishing, LLC.
2. Zheng, A., & Casari, A. (2018). *Feature engineering for machine learning: principles and techniques for data scientists*. " O'Reilly Media, Inc."
3. Kuhn, M., & Johnson, K. (2019). *Feature engineering and selection: A practical approach for predictive models*. Chapman and Hall/CRC.
4. Kimball, R., & Ross, M. (2013). *The data warehouse toolkit: The definitive guide to dimensional modeling*. John Wiley & Sons.
5. White, T. (2012). *Hadoop: The definitive guide*. " O'Reilly Media, Inc."
6. Chambers, B., & Zaharia, M. (2018). *Spark: The definitive guide: Big data processing made simple*. " O'Reilly Media, Inc."
7. Reis, J., & Housley, M. (2022). *Fundamentals of data engineering*. " O'Reilly Media, Inc.".
8. Domingos, P. (2012). A few useful things to know about machine learning. *Communications of the ACM*, *55*(10), 78-87.
9. Kanter, J. M., & Veeramachaneni, K. (2015, October). Deep feature synthesis: Towards automating data science endeavors. In *2015, the IEEE International Conference on Data Science and Advanced Analytics (DSAA)* (pp. 1-10). IEEE.
10. Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. *Communications of the ACM*, *51*(1), 107-113.
11. Armbrust, M., Das, T., Sun, L., Yavuz, B., Zhu, S., Murthy, M., ... & Zaharia, M. (2020). Delta lake: high-performance ACID table storage over cloud object stores. *Proceedings of the VLDB Endowment*, *13*(12), 3411-3424.
12. Polyzotis, N., Roy, S., Whang, S. E., & Zinkevich, M. (2017, May). Data management challenges in production machine learning. In *Proceedings of the 2017 ACM International Conference on Management of Data* (pp. 1723-1726).
13. Atwal, H. (2020). Practical DataOps. *Practical DataOps (1st ed.). Apress Berkeley, CA. https://doi. org/10.1007/978-1-4842-5104-1*.
14. Mainali, K. (2020). DataOps: Towards Understanding and Defining Data Analytics Approach.
15. E. Breck et al., “The ML test score: A rubric for ML production readiness and technical debt reduction,” in *Proc. IEEE Int. Conf. Big Data*, 2017, pp. 1123–1132.
16. Demchenko, Y., Cuadrado-Gallego, J. J., Chertov, O., & Aleksandrova, M. (2024). Data science projects management, dataops, mlops. In *Big data infrastructure technologies for data analytics: scaling data science applications for continuous growth* (pp. 447-497). Cham: Springer Nature Switzerland.
17. Akidau, T., Chernyak, S., & Lax, R. (2018). *Streaming systems: the what, where, when, and how of large-scale data processing*. " O'Reilly Media, Inc."
18. Kumar, N. (2026). Cyber Security Threats and Challenges in CPS. *AI and Cyber Security in Cyber-Physical Systems*, 53-78.
19. Armbrust, M., Ghodsi, A., Xin, R., & Zaharia, M. (2021, January). Lakehouse: a new generation of open platforms that unify data warehousing and advanced analytics. In *Proceedings of CIDR* (Vol. 8, No. 1, p. 28). sn.
20. Salvucci, E. (2021). MLOps-Standardizing the Machine Learning Workflow.