International Journal of AI, Big Data, Computational and Management Studies
Noble Scholar Research Group | Volume 7, Issue 2, 85-91, 2026
ISSN: 3050-9416 | https://doi.org/10.63282/3050-9416.IJAIBDCMS-V7I2P115

Original Article

Integrating Generative AI with Real-Time Data Pipelines for Operational Decision Intelligence

Stewyn Chaudhary
Independent Researcher, USA.

Received On: 23/02/2026 ***Revised On: 02/04/2026*** ***Accepted On: 09/04/2026*** ***Published On: 16/04/2026***

Abstract**:** *This paper presents a comprehensive conceptual framework for integrating generative artificial intelligence (AI) with real-time data pipelines to enable operational decision intelligence. As organizations increasingly adopt AI-driven solutions, the ability to leverage real-time data streams with generative AI models becomes crucial for timely, informed decision-making. We propose a multi-layered architecture that encompasses data ingestion, AI processing, and decision intelligence generation, complemented by continuous feedback loops for model improvement. The framework addresses key challenges including latency management, model deployment strategies, and ethical considerations in AI-driven operations. Through examination of existing literature and emerging best practices, we demonstrate how this integration can enhance operational agility and decision quality across diverse domains. The paper discusses implementation considerations, trade-offs between performance metrics, and governance requirements. Finally, we outline future research directions including advanced interpretability methods, federated learning approaches, and multi-model orchestration strategies. This work contributes to the growing body of knowledge on AI systems architecture and provides practitioners with actionable insights for building intelligent operational systems.*

Keywords: *Generative AI, real-time data pipelines, decision intelligence, streaming architecture, AI operations, machine learning engineering, data-driven decision making.*

1. Introduction

The convergence of generative artificial intelligence (GenAI) and real-time data processing represents a significant paradigm shift in how organizations make operational decisions. Generative AI models, particularly large language models (LLMs) and transformer-based architectures, have demonstrated remarkable capabilities in understanding context, generating insights, and producing actionable recommendations [1], [2]. Simultaneously, advances in distributed streaming technologies such as Apache Kafka, Apache Flink, and cloud-native solutions have enabled organizations to process vast data streams with sub-second latencies [3], [4]. However, the effective integration of these two capabilities—leveraging real-time data to inform generative AI models that drive operational decisions—remains largely unexplored in academic literature, despite its growing practical importance.

Decision intelligence represents the intersection of data science, domain expertise, and business acumen, focused on translating data insights into actionable decisions [5], [6]. Traditional decision support systems have relied on rule-based engines or statistical models that require substantial human interpretation. The integration of generative AI introduces new possibilities: AI systems can synthesize complex data patterns, generate natural language explanations, propose multiple decision scenarios, and adapt recommendations in real time based on evolving data streams [7], [8]. This capability is particularly valuable in high-frequency, dynamic operational environments such as financial trading, supply chain optimization, predictive maintenance, and fraud detection, where decision latency directly impacts organizational outcomes.

Despite these developments, significant technical and organizational challenges persist. First, the integration of generative AI with real-time systems introduces complexity in ensuring response latency meets operational requirements while maintaining model accuracy and relevance [9]. Second, the dynamic nature of real-time data streams presents challenges for prompt engineering, retrieval-augmented generation (RAG) systems, and context management in AI models [10]. Third, governance, ethics, and explainability concerns become more acute when AI-driven decisions operate at operational speeds without human intervention [11], [12]. This paper addresses these gaps by proposing a comprehensive conceptual framework that integrates generative AI with real-time data pipelines for decision intelligence, while explicitly addressing latency management, model governance, and ethical considerations.

The primary contribution of this work is a multi-layered reference architecture that demonstrates how generative AI can be effectively integrated into real-time data processing pipelines to generate actionable decision intelligence. We examine the technical requirements at each layer, discuss implementation trade-offs, and provide practical guidance for practitioners. The framework is domain-agnostic,

designed to be applicable across various operational scenarios including supply chain management, financial services, healthcare operations, and manufacturing systems.

2. Literature Review

2.1. Generative AI in Enterprise Systems

Generative AI has transitioned from academic research to practical enterprise deployment in recent years. Large language models like GPT-4, Claude, and domain-specific models have demonstrated effectiveness in various business applications including customer service, content generation, and technical documentation [1], [13]. Beyond text generation, generative models now encompass multimodal capabilities, enabling integration of text, images, and structured data [2]. The enterprise adoption of generative AI has been driven by improvements in model capabilities, the emergence of accessible APIs, and the development of operational frameworks for responsible AI deployment [14], [15]. However, deploying generative AI in enterprise systems presents distinct challenges compared to traditional ML models. Issues such as hallucination, context window limitations, non-deterministic outputs, and computational cost have led to the development of complementary techniques.

Retrieval-Augmented Generation (RAG) has emerged as a critical pattern for grounding generative models in authoritative data sources [10], [16]. Prompt engineering, few-shot learning, and fine-tuning approaches have become standard practices for optimizing model performance for specific enterprise use cases [17]. Additionally, orchestration frameworks and MLOps practices for generative AI are still evolving, with emerging standards for model serving, versioning, and governance [18].

2.2. Real-Time Data Pipeline Architectures

Real-time data processing has matured significantly with the emergence of stream processing frameworks and cloud-native architectures. Apache Kafka has become the de facto standard for building scalable, fault-tolerant event streaming platforms, supporting organizations that process trillions of events daily [3], [19]. Apache Flink, Spark Streaming, and cloud-native solutions like AWS Kinesis and Google Pub/Sub have extended the ecosystem with sophisticated stateful processing capabilities [4], [20]. These technologies enable organizations to build real-time data pipelines that combine event streaming, transformations, aggregations, and enrichment with sub-second latencies.

Modern data pipeline architectures emphasize decoupling, scalability, and operational resilience. The Lambda and Kappa architectures have provided foundational patterns for integrating batch and real-time processing [21]. Microservices approaches to data pipeline design have increased flexibility and independent scalability of pipeline components [22]. However, integrating complex AI models, particularly computationally expensive generative models, into real-time pipelines presents new challenges. The latency requirements for model inference, the computational resources required, and the need to manage model state alongside streaming data state are active areas of research [23], [24].

2.3. Decision Intelligence Paradigms

Decision intelligence represents an evolution beyond traditional business intelligence and analytics. While business intelligence focuses on descriptive and diagnostic analytics (what happened and why), decision intelligence emphasizes prescriptive and predictive analytics aimed at informing and automating decisions [5], [25]. The field integrates insights from organizational decision science, behavioral economics, and advanced analytics to create systems that not only generate insights but actively support decision-making processes [6].

Recent developments in decision intelligence have emphasized human-AI collaboration models, where AI systems augment rather than replace human decision-makers [26], [27]. This paradigm recognizes that effective operational decisions require both data-driven insights and human judgment, particularly in complex, novel, or high-stakes scenarios. Research has also highlighted the importance of explainability and transparency in decision intelligence systems, with regulatory frameworks increasingly requiring organizations to provide interpretable explanations for AI-driven decisions [11], [28]. The integration of real-time data with decision intelligence further emphasizes the need for responsive, adaptive systems that can adjust recommendations as conditions change [12].

3. Proposed Conceptual Framework

3.1. Framework Overview

We propose a multi-layered architecture that systematically integrates generative AI with real-time data pipelines to generate operational decision intelligence. The framework consists of five interconnected layers: (1) Data Ingestion and Streaming, (2) Data Processing and Enrichment, (3) Generative AI Processing, (4) Decision Intelligence Generation, and (5) Feedback and Continuous Learning [1], [5]. Each layer serves a distinct function while maintaining clear interfaces with adjacent layers, enabling modularity, scalability, and independent evolution of components. This architectural approach allows organizations to implement the framework progressively, beginning with foundational real-time data capabilities and gradually incorporating AI-driven decision generation.

The framework is designed around several key principles. First, separation of concerns ensures that data pipeline logic, AI model logic, and decision logic remain independently managed and evolved [3], [4]. Second, event-driven architecture enables asynchronous processing, reducing latency-critical path dependencies and improving system resilience [19], [20].

Third, immutability and event-sourcing principles provide audit trails for compliance and enable reconstruction of system state [21]. Fourth, polyglot persistence allows specialized technologies to handle different aspects streaming brokers for event ingestion, vector databases for

RAG systems, and model registries for AI model management [10], [22]. Finally, human-in-the-loop decision mechanisms ensure that AI recommendations are subject to appropriate oversight and governance [26], [27].

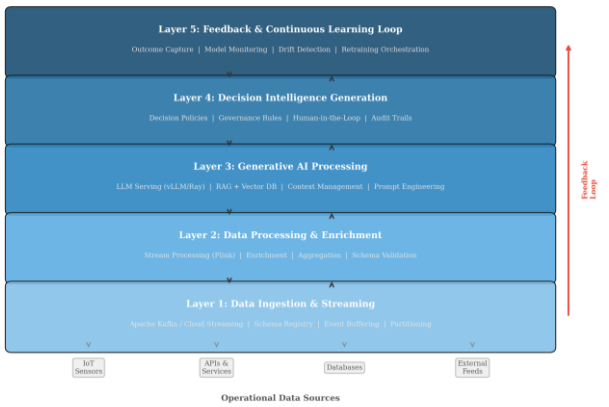

Fig 1: Multi-Layered Architecture for GenAI-Driven Decision Intelligence.

3.2. Data Ingestion and Streaming Layer

The foundation of the framework is the Data Ingestion and Streaming Layer, responsible for reliably capturing, buffering, and distributing event streams from diverse operational sources. This layer employs distributed message brokers such as Apache Kafka or cloud-native streaming services that provide durable, fault-tolerant storage of events [3], [19]. A critical design consideration is the definition of event schemas and the establishment of schema governance to ensure data quality and enable downstream systems to evolve independently [29]. Events are typically immutable records capturing what happened in the operational system, with timestamps indicating when events occurred and when they were observed by the system.

Beyond simple event capture, this layer must handle challenges including backpressure management (ensuring producers do not overwhelm the system), late-arriving events, out-of-order delivery, and event deduplication [4], [20]. The layer also includes capability for topic partitioning to enable parallel processing and scaling to high-throughput scenarios. For many operational domains, the Data Ingestion Layer must also manage historical data retention, enabling systems to process both real-time streams and backfilled historical data for model training and validation [23], [30]. This unified approach to batch and real-time data ingestion aligns with the modern Kappa architecture paradigm [21].

3.3. Data Processing and Enrichment Layer

The Data Processing and Enrichment Layer sits between raw data ingestion and the generative AI components, performing essential transformations that prepare streaming data for downstream AI consumption. This layer leverages stream processing engines such as Apache Flink, Spark Streaming, or cloud-native equivalents to execute real-time transformations including filtering, aggregation, windowing, and joins across multiple event streams [4], [20]. A key function of this layer is data enrichment augmenting raw events with contextual information from reference data stores, historical databases, and external APIs to create semantically rich event payloads suitable for AI model consumption [3], [29].

Schema validation and data quality enforcement are critical responsibilities at this layer. Malformed, incomplete, or anomalous events must be identified and handled before reaching the AI processing components, as generative models are sensitive to input quality [18], [29]. The layer implements stateful processing patterns—maintaining session windows, counting aggregates, and detecting complex event patterns that provide higher-level abstractions over raw event streams [4]. Additionally, this layer manages feature computation for AI models, computing real-time features from streaming data that complement batch-computed features from the data warehouse, following a feature store pattern that ensures consistency between training and inference environments [23], [24].

3.4. Generative AI Processing Layer

The Generative AI Processing Layer encapsulates the logic for applying generative models to enrich events and generate contextual insights from streaming data. Rather than applying generative models directly to every event which would create latency and cost challenges this layer implements intelligent filtering and batching strategies [9]. Events relevant to specific decision domains are routed to appropriate generative models, while computational resources are shared and optimized through model serving infrastructures such as vLLM, Ray Serve, or Seldon Core [24], [31].

A critical component of this layer is the context management system, which maintains relevant historical and domain context for AI models. As generative models operate within fixed context windows, the system must dynamically select the most relevant information from recent events, historical data, and reference materials [10], [16]. This often employs retrieval-augmented generation (RAG) patterns where vector databases store embeddings of historical data, enabling semantic search for contextually relevant information [10]. Prompt engineering strategies are applied at this layer to structure model inputs in ways that elicit high-quality, decision-relevant outputs [17], [32].

The layer also implements fallback mechanisms and quality assurance. If a generative model fails, times out, or produces low-confidence outputs, the system can degrade gracefully to rule-based fallbacks or previous recommendations [33]. Model monitoring at this layer tracks output quality, drift in model behavior, and cost metrics to inform model selection and retraining decisions [18], [34]. The layer maintains versioning of prompts, models, and retrieval strategies, enabling A/B testing and progressive rollout of AI model improvements.

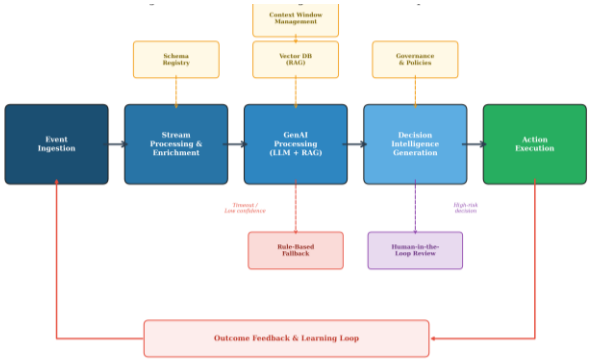

Fig 2: Data Flow from Event Ingestion to Decision Output.

3.5. Decision Intelligence Layer

Building on insights from the generative AI processing layer, the Decision Intelligence Layer synthesizes recommendations, evaluates alternatives, and generates actionable decision guidance. This layer receives structured outputs from generative models and applies decision logic specific to the operational domain [5], [6]. For example, in a supply chain optimization scenario, the Decision Intelligence Layer might receive AI-generated demand forecasts, supplier availability assessments, and logistics recommendations, then synthesize these into procurement and inventory decisions [25].

A critical responsibility of this layer is ensuring that decisions comply with business rules, regulatory requirements, and governance policies [11], [12]. The layer implements decision policies that specify thresholds, constraints, and escalation procedures. For high-impact decisions or situations where confidence is low, the system routes decisions for human review, implementing human-in-the-loop workflows [26], [27]. The layer also tracks decision reasoning—maintaining an audit trail of what data, what model outputs, and what decision policies led to each recommendation [28]. This explainability capability is essential for regulatory compliance and building organizational trust in AI-driven decisions.

The Decision Intelligence Layer must also manage the temporal aspects of decisions. Some decisions have immediate effect and should be executed upon generation, while others require scheduling or sequencing with other operational actions [29]. The layer coordinates with operational systems to execute decisions, monitor their execution, and capture feedback on decision outcomes for the learning loop discussed below.

3.6. Feedback and Continuous Learning Loop

Recognizing that static models and decision policies become suboptimal as operational conditions evolve, the framework includes a comprehensive feedback and continuous learning loop [35]. Outcomes of decisions both intended results and unexpected consequences are captured as new events returning to the Data Ingestion Layer [30]. These outcome events are joined with the initial data and decisions that prompted them, creating rich training data for model improvement [23], [24]. The learning loop operates at multiple timescales. Short-term feedback (minutes to hours) informs real-time model adjustments and prompt engineering refinements [17], [32]. Medium-term feedback (days to weeks) drives retraining decisions and evaluation of model alternatives [18], [34]. Long-term feedback (months to years) informs strategic decisions about architecture changes, introduction of new data sources, and evolution of decision policies [35]. This multi-timescale approach acknowledges that different types of learning require different observation windows and computational resources [31]. The framework includes automated model evaluation, drift detection, and retraining orchestration to minimize human operational overhead while maintaining system performance [34], [36].

4. Implementation Considerations

4.1. Latency and Throughput Trade-offs

A primary consideration when implementing the framework is managing the trade-off between response latency and system throughput. Generative models, particularly large models, incur significant computational cost and latency—typically ranging from 100 milliseconds to several seconds per inference [9], [24]. For operational decision systems, acceptable latency depends on the domain: high-frequency trading may require sub-100-millisecond decisions, while supply chain optimization might tolerate seconds of latency [23]. Organizations must explicitly define latency service level agreements (SLAs) and architect systems accordingly [37].

Several design patterns reduce latency. Model quantization (reducing precision) and knowledge distillation (training smaller models to mimic larger ones) can reduce inference latency while maintaining acceptable quality [38]. Batching requests across multiple events amortizes model loading overhead, though it introduces queuing delays [9]. Caching of model outputs for similar inputs reduces redundant computation. Hybrid approaches that use fast rule-based systems for common cases and deploy generative models only when needed can optimize the overall latency distribution [33]. Ultimately, many implementations employ tiered decision strategies: fast approximate decisions immediately, higher-quality decisions within tolerable latency windows, and offline batch processing for non-urgent decisions [37], [39].

4.2. Model Selection and Deployment

Organizations face choices regarding which generative models to employ. Commercial API-based models (such as OpenAI's GPT-4 or Anthropic's Claude) offer state-of-the-art capabilities but introduce vendor lock-in and ongoing operational costs that scale with usage [1], [2]. Open-source models (Llama, Mistral, and others) provide greater control and can be deployed locally but require substantial infrastructure and expertise to optimize [40]. Fine-tuned or specialized models may provide better performance for specific domains but require significant training data and computational resources [17], [18].

Deployment strategies significantly impact operational characteristics. Running models on GPU clusters provides high throughput but requires capital investment and operational complexity. Serverless inference platforms (AWS SageMaker, Google Vertex AI) provide elasticity and reduce operational burden at the cost of higher per-request costs and less control over resource allocation [31]. Multi-model strategies, where different model sizes are selected based on request complexity, can optimize cost-quality trade-offs [24], [38]. Canary deployments and gradual rollouts enable testing of new models before full production deployment, reducing risk of degraded decision quality [18], [36].

4.3. Ethical and Governance Considerations

The deployment of generative AI systems for operational decision-making raises significant ethical and governance concerns. Bias in model outputs, particularly when trained on data reflecting historical inequities, can perpetuate or amplify discrimination [11], [41]. Explainability challenges the difficulty in understanding why generative models produce specific outputs complicate efforts to ensure decisions are made fairly and transparently [28], [42].

Governance frameworks for AI-driven decision systems must address several dimensions [12], [43]. Accountability structures should clearly define responsibility for decisions made by AI systems. Transparency mechanisms should enable stakeholders to understand what data, what models, and what decision policies influenced specific decisions. Bias monitoring should continuously assess whether decision outcomes differ across demographic groups or business segments [44]. Human oversight mechanisms including human-in-the-loop review, escalation procedures, and decision reversal capabilities—should be proportional to decision impact and risk [27]. Organizations must also address data privacy concerns, ensuring that personal data used in decision-making is handled in compliance with regulations such as GDPR and used only for legitimate purposes [45]. Finally, intellectual property considerations arise regarding rights to model outputs and responsibility for decisions informed by third-party models [46].

5. Discussion

The integration of generative AI with real-time data pipelines represents a significant evolution in operational systems. Theoretically, the framework contributes to understanding how AI capabilities can enhance decision-making in complex, dynamic environments by systematically addressing the integration of multiple technical and organizational layers [5], [6]. The framework formalizes the multi-timescale learning processes necessary for AI systems to improve as operational conditions evolve [35]. It also emphasizes the centrality of human oversight and explainability, contributing to the growing literature on responsible AI and human-AI collaboration [26], [27], [28]. Practically, the framework provides a blueprint for organizations implementing AI-driven operational decision systems. By decomposing the integration into distinct layers with clear responsibilities and interfaces, the framework enables modular implementation and progressive value realization. Organizations need not implement all components simultaneously; they can begin with foundational streaming infrastructure and progressively add AI capabilities [3], [4]. The framework's emphasis on feedback loops and continuous learning directly addresses a critical gap in many AI implementations, which often treat models as static artifacts rather than continuously evolving systems [35], [36].

Compared with existing approaches, the framework advances the state of practice in several ways. Traditional business intelligence dashboards provide insights but require human interpretation to translate insights into decisions; the framework automates decision generation through generative AI [5]. Rule-based decision engines provide deterministic, explainable decisions but lack adaptability and semantic understanding; the framework combines rule-based and AI-driven approaches [33]. Recent machine learning operations (MLOps) frameworks have focused on managing model lifecycle but often sideline the integration of models with operational decision logic; the framework explicitly addresses this integration [18], [34]. Emerging decision intelligence platforms have emphasized the synthesis of multiple data sources and recommendation generation but often operate on batch schedules; the framework emphasizes real-time responsiveness [6], [25]. Finally, while human-AI collaboration literature has addressed specific techniques, the framework provides an integrated architectural approach to operationalizing human-AI collaboration at scale [26], [27].

Limitations of the framework merit discussion. The framework is intentionally domain-agnostic, which provides generality but may overlook important domain-specific considerations. Real-time decision-making in critical domains such as healthcare or autonomous systems requires domain-specific safety mechanisms beyond the general governance considerations discussed [11], [12]. The framework emphasizes ideal-case architectures but does not comprehensively address resource constraints or organizational realities in many enterprises, where full implementation of all layers may be infeasible. Additionally, the framework does not address specific technical challenges in emerging areas such as federated learning across distributed organizations or handling adversarial data and model attacks [47], [48].

Tables 1: Comparison of Proposed Framework with Existing Approaches

Approach	Key Strength	Key Limitation	Real-Time AI Integration
Business Intelligence	Comprehensive data views	Requires human interpretation	No AI integration
Rule-Based Systems	Deterministic, explainable	Lacks adaptability	Limited semantic

			understanding
Traditional ML Pipelines	Proven methodologies	Requires labeled data	Separate from decision logic
Decision Intelligence Platforms	Multi-source synthesis	Often batch-oriented	Limited generative AI
Proposed Framework	Real-time GenAI with governance	Requires significant infrastructure	Full integration with feedback loops

6. Conclusion and Future Directions

This paper has presented a comprehensive conceptual framework for integrating generative AI with real-time data pipelines to generate operational decision intelligence. The multi-layered architecture, spanning data ingestion, AI processing, decision generation, and continuous learning, provides both a theoretical contribution to understanding AI systems architecture and practical guidance for implementation. By explicitly addressing latency management, model governance, ethical considerations, and human oversight, the framework advances the maturity of AI systems beyond isolated model optimization toward holistic operational systems that create business value while managing risk [1], [5], [11]. Future research directions include several promising areas. First, advanced interpretability and explainability methods are needed to make generative model decisions more transparent, particularly for high-stakes domains [42], [43]. Second, federated learning and privacy-preserving AI techniques are essential for enabling organizations to collaboratively train and improve models without exposing sensitive data [47]. Third, multi-model orchestration and dynamic model selection strategies can optimize cost-quality-latency trade-offs [38], [39]. Fourth, formal verification and robustness testing methods are needed to ensure AI-driven operational systems are resilient to adversarial inputs and model failures [48]. Finally, empirical case studies documenting implementation experiences, lessons learned, and measurable business outcomes would significantly strengthen the foundation for practice [49].

References

1. S. Bubeck, V. Chandrasekaran, R. Eldan, et al., “Sparks of artificial general intelligence: Early experiments with GPT-4,” arXiv preprint arXiv:2303.12712, 2023.
2. C. Raffel, N. Shazeer, A. Roberts, et al., “Exploring the limits of transfer learning with a unified text-to-text transformer,” J. Mach. Learn. Res., vol. 21, no. 140, pp. 1–67, 2020.
3. J. Kreps, N. Narkhede, and J. Rao, “Kafka: A distributed messaging system for log processing,” in Proc. NetDB Workshop, Athens, Greece, 2011.
4. P. Carbone, A. Katsifodimos, S. Ewen, et al., “Apache Flink: Stream and batch processing in a single engine,” IEEE Data Eng. Bull., vol. 38, no. 4, pp. 28–38, 2015.
5. L. Pratt, Link: How Decision Intelligence Connects Data, Actions, and Outcomes for a Better World. Hoboken, NJ, USA: Wiley, 2022.
6. P. Leonelli, D. Hutter, D. Melnick, and P. Rensing, “The missing link: What machine learning needs from operations,” McKinsey Analytics, 2021.
7. Y. Bai, S. Kadavath, S. Kundu, et al., “Constitutional AI: Harmlessness from AI feedback,” arXiv preprint arXiv:2212.08073, 2022.
8. Y. Zhou, A. I. Mazzoni, Y. Tay, et al., “Large language models as zero-shot planners for task management,” arXiv preprint, 2023.
9. J. M. Hellerstein, C. Ré, F. Schoppmann, et al., “The MADlib analytics library or MAD skills, the SQL,” Proc. VLDB Endowment, vol. 5, no. 12, pp. 1700–1711, 2012.
10. P. Lewis, E. Perez, A. Piktus, et al., “Retrieval-augmented generation for knowledge-intensive NLP tasks,” in Advances in Neural Inf. Process. Syst., vol. 33, pp. 9459–9474, 2020.
11. B. Mittelstadt, “From individual to group privacy in big data analytics,” Philosophy & Technology, vol. 30, no. 4, pp. 475–494, 2017.
12. F. Brynielsson, A. Horndahl, L. Kaati, et al., “Harvesting and analyzing web data for security applications,” in Proc. IEEE Int. Conf. Intelligence and Security Informatics, 2013.
13. A. Vaswani, N. Shazeer, N. Parmar, et al., “Attention is all you need,” in Advances in Neural Inf. Process. Syst., vol. 30, 2017.
14. D. Ganguli, L. Lovitt, J. Kernion, et al., “Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned,” arXiv preprint arXiv:2209.07858, 2022.
15. S. Shankar, R. Garcia, J. M. Hellerstein, and A. G. Parameswaran, “Operationalizing machine learning: An interview study,” arXiv preprint arXiv:2209.09125, 2022.
16. H. Ren, K. Zuo, J. Tang, and M. Coates, “Improving retrieval augmented language models by searching in-context examples,” in Proc. EMNLP, 2023.
17. B. Zhao, Y. Qi, Z. Yuan, et al., “A survey of the research on large language model prompt engineering,” arXiv preprint, 2023.
18. D. Sculley, G. Holt, D. Golovin, et al., “Hidden technical debt in machine learning systems,” in Advances in Neural Inf. Process. Syst., 2015.
19. T. Akidau, S. Bradshaw, C. Chambers, et al., “The Dataflow model: A practical approach to balancing correctness, latency, and cost in massive-scale, unbounded, out-of-order data processing,” Proc. VLDB Endowment, vol. 8, no. 12, pp. 1792–1803, 2015.
20. G. Hesse, C. Matthies, K. Perscheid, M. Uflacker, and H. Plattner, “Quantitative impact evaluation of an abstraction layer for data stream processing systems,” in Proc. IEEE 39th Int. Conf. Distributed Computing Systems (ICDCS), Dallas, TX, USA, pp. 1381–1392, 2019.

21. J. Kreps, "Questioning the Lambda architecture," O'Reilly Architecture Summit, 2014.
22. N. Newman, Microservices in Action. Manning Publications, 2021.
23. J. Dean and S. Ghemawat, "MapReduce: Simplified data processing on large clusters," Commun. ACM, vol. 51, no. 1, pp. 107–113, 2008.
24. A. Paszke, S. Gross, F. Massa, et al., "PyTorch: An imperative style, high-performance deep learning library," in Advances in Neural Inf. Process. Syst., vol. 32, 2019.
25. S. Rai, "Supply chain 4.0: Digital transformation of supply chain management," J. Enterprise Inf. Management, vol. 34, no. 1, pp. 1–32, 2020.
26. S. Amershi, D. Weld, M. Vorvoreanu, et al., "Guidelines for human-AI interaction," in Proc. 2019 CHI Conf. Human Factors in Computing Systems (CHI '19), 2019.
27. B. Green and M. Banfield, "Artificial intelligence and high-risk decisions: Assessment framework and guideline," Data & Society Research Institute, 2022.
28. M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in Proc. 22nd ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining, 2016.
29. M. Stonebraker, "The case for polystores," ACM SIGMOD Record, vol. 44, no. 3, pp. 33–40, 2015.
30. D. Maier, "Temporal databases: From theory to practice," in Temporal Databases: Research and Practice. Springer, 1998, pp. 1–31.
31. M. Zaharia, M. Chowdhury, M. J. Franklin, S. Shenker, and I. Stoica, "Spark: Cluster computing with working sets," in Proc. 2nd USENIX Workshop Hot Topics Cloud Comput. (HotCloud'10), Boston, MA, 2010.
32. T. Brown, B. Mann, N. Ryder, et al., "Language models are few-shot learners," in Advances in Neural Inf. Process. Syst., vol. 33, 2020.
33. A. Paleyes, R. G. Urma, and N. D. Lawrence, "Challenges in deploying machine learning: A survey of case studies," ACM Computing Surveys, vol. 55, no. 6, pp. 1–29, 2022.
34. K. Yang, J. Tian, N. Katariya, et al., "Machine learning observability: A framework for improving transparency and trust in ML/AI systems," in Proc. IEEE Int. Conf. Data Mining (ICDM), 2021.
35. J. Pearl, Causality: Models, Reasoning, and Inference, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 2009.
36. D. Chen, Y. Chen, X. Shi, and B. Wang, "AutoML: A survey of the state-of-the-art," Knowledge-Based Systems, vol. 212, p. 106622, 2021.
37. P. Carbone, G. Fögen, S. Ewen, et al., "Lightweight asynchronous snapshots for distributed dataflows," arXiv preprint arXiv:1506.08603, 2015.
38. M. Tan, R. Pang, and Q. V. Le, "EfficientNet: Rethinking model scaling for convolutional neural networks," in Proc. Int. Conf. Machine Learning (ICML), PMLR, 2019.
39. C. Liu, X. Chen, Y. Zhou, et al., "Resource efficient machine learning in 2 KB RAM for the Internet of Things," in Proc. IEEE 17th Int. Conf. Data Mining (ICDM), 2017.
40. T. Dettmers, M. Lewis, S. Belkada, and L. Zettlemoyer, "LLM.int8(): 8-bit matrix multiplication for transformers at scale," in Advances in Neural Inf. Process. Syst., vol. 35, pp. 30326–30339, 2022.
41. J. Buolamwini and T. Gebru, "Gender shades: Intersectional accuracy disparities in commercial gender classification," in Proc. Conf. Fairness, Accountability and Transparency (FAT), 2018.
42. S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in Advances in Neural Inf. Process. Syst., pp. 4765–4774, 2017.
43. F. Doshi-Velez and B. Kim, "Towards a rigorous science of interpretable machine learning," arXiv preprint arXiv:1702.08608, 2017.
44. A. Lambrecht and C. Tucker, "Algorithmic bias? An empirical study of apparent gender-based discrimination in the display of STEM career ads," Management Science, vol. 65, no. 7, pp. 2966–2981, 2019.
45. S. Wachter, B. Mittelstadt, and L. Floridi, "Why a right to explanation of automated decision-making does not exist in the GDPR," Int. Data Privacy Law, vol. 7, no. 2, pp. 76–99, 2017.
46. J. H. Reichman and J. C. Ginsburg, "The Berlin Declaration on open access to knowledge in the sciences and humanities," SSRN Electronic J., 2004.
47. P. Kairouz, B. McMahan, B. Avent, et al., "Advances and open problems in federated learning," Foundations and Trends in Machine Learning, vol. 14, no. 1–2, pp. 1–210, 2021.
48. A. Ilyas, S. Santurkar, D. Tsipras, L. Engstrom, B. Tran, and A. Madry, "Adversarial examples are not bugs, they are features," in Advances in Neural Inf. Process. Syst., vol. 32, pp. 125–136, 2019.
49. W. J. Orlikowski and C. S. Iacono, "Research commentary: Desperately seeking the 'IT' in IT research—a call to theorizing the IT artifact," Inf. Systems Research, vol. 12, no. 2, pp. 121–134, 2001.