Transitioning from MLOps to LLMOps: Navigating the Unique Challenges of Large Language Models

Article *in* Information · January 2025

DOI: 10.3390/info16020087

information

Review

Transitioning from MLOps to LLMOps: Navigating the Unique Challenges of Large Language Models

Saurabh Pahune [1] and **Zahid Akhtar** [2,*]

[1] Cardinal Health, Dublin, OH 43017, USA; saurabh.pahune@cardinalhealth.com
[2] Department of Network and Computer Security, State University of New York Polytechnic Institute, Utica, NY 13502, USA
* Correspondence: akhtarz@sunypoly.edu; Tel.: +1-315-792-7238

Abstract: Large Language Models (LLMs), such as the GPT series, LLaMA, and BERT, possess incredible capabilities in human-like text generation and understanding across diverse domains, which have revolutionized artificial intelligence applications. However, their operational complexity necessitates a specialized framework known as LLMOps (Large Language Model Operations), which refers to the practices and tools used to manage lifecycle processes, including model fine-tuning, deployment, and LLMs monitoring. LLMOps is a subcategory of the broader concept of MLOps (Machine Learning Operations), which is the practice of automating and managing the lifecycle of ML models. LLM landscapes are currently composed of platforms (e.g., Vertex AI) to manage end-to-end deployment solutions and frameworks (e.g., LangChain) to customize LLMs integration and application development. This paper attempts to understand the key differences between LLMOps and MLOps, highlighting their unique challenges, infrastructure requirements, and methodologies. The paper explores the distinction between traditional ML workflows and those required for LLMs to emphasize security concerns, scalability, and ethical considerations. Fundamental platforms, tools, and emerging trends in LLMOps are evaluated to offer actionable information for practitioners. Finally, the paper presents future potential trends for LLMOps by focusing on its critical role in optimizing LLMs for production use in fields such as healthcare, finance, and cybersecurity.

Keywords: large language models (LLMs); LLMOps; MLOps; model fine-tuning; infrastructure scalability; ethical AI practices; security in AI operations; generative AI (GenAI); LangChain; Vertex AI; retrieval-augmented generation (RAG); GPT; text generation; cybersecurity

check for updates

Academic Editor: Rodolfo Delmonte

Received: 30 December 2024
Revised: 15 January 2025
Accepted: 19 January 2025
Published: 23 January 2025

Citation: Pahune, S.; Akhtar, Z. Transitioning from MLOps to LLMOps: Navigating the Unique Challenges of Large Language Models. *Information* **2025**, *16*, 87. https://doi.org/10.3390/info16020087

1. Introduction

Artificial intelligence (AI) and machine learning (ML) have made substantial progress in recent years. They have transformed various industries and are influencing daily life, enabling companies to accelerate progress in decision making and strategic planning [1]. Generative artificial intelligence (GenAI) has recently been implemented in numerous businesses and areas to transform traditional practices. The application of advanced AI technologies has impacts on sectors like manufacturing [2], cybersecurity [3], medicine and healthcare [4,5], supply chain management [6,7], software development [8], air transport industry [9], and everyday software automation, and is widespread in several industries [10,11]. Despite remarkable progress, many potential ML applications have struggled to meet expectations in real-world scenarios and failed or faced challenges to be deployed in production. Understanding the various challenges is crucial for improving the success rate of ML deployment. These challenges include unqualified data collection practices or biased datasets [12], data security and privacy issues such as ML model deployments that can cause leakage of private information [13–15], ML models experiencing delays in processing or inference—which can affect applications that run in real time due to the complexity size of the ML model [16]—and drift issues—which refer to changes in data or model behavior over time and critically affect ML model performance in production [17].

To effectively address the challenges connected to the deployment of ML models in production, it is necessary to analyze the current research focus and explore the utilization of MLOps. This approach optimizes the procedure for the deployment, management, and supervision of machine learning models in a production environment [18]. It requires years of collaboration among data scientists, front-end developers, machine learning engineers, and production engineers. These teams must work together to share their expertise and establish efficient workflows for transitioning models into production. However, the process is fraught with challenges that are among the primary reasons why only a small fraction of ML projects successfully reach production [19].

Advancements in LLMs (e.g., BERT, GPT, LLaMA, and T5) have revolutionized ML by enabling human-like text generation and understanding [20–23]. These models are applied in various domains, such as financial LLMs based on tasks [24], multilingual LLMs [25], clinical and biomedical LLMs [26], vision language LLMs [27] and code language models [28]. But they face challenges like high computational requirements and the need for fine-tuning [29]. A critical concern in managing LLMs is ensuring their longevity and avoiding model drift, which can affect their performance over time [30]. Lifelong learning is vital to enable LLMs to adapt to evolving data, tasks, and user preferences, as static datasets are inadequate to handle the dynamic nature of real-world information [31]. Cybersecurity risks further underscore the importance of monitoring and securing LLMs against adversarial threats [32,33]. Adversaries are increasingly employing LLMs to carry out a range of attacks. Examples include backdoor attacks [34] (i.e., unauthorized access is secretly gained that enables manipulation or extraction of sensitive data from targeted systems), transfer-based black-box attacks [35,36] (i.e., adversarial examples using a surrogate model that can be transferred to a target model without access to internal structure), data-poisoning attacks [37] (i.e., maliciously injecting malicious data into the training set to degrade model performance or induce biased, incorrect, or harmful outputs), and jailbreak attacks [38] (i.e., exploiting vulnerabilities in the model using malicious prompts to bypass safety restrictions and make the model behave in unintended ways, e.g., generating prohibited or harmful content).

In total, the complexity of LLMs, which involves billions of parameters and vast datasets, demands robust lifecycle management [39]. To address the challenges in LLMs, LLMOps (Large Language Model Operations) provides tools for efficient data handling, model training, deployment, and maintenance, addressing issues such as model drift and ensuring adaptability to changing data and tasks [40]. Essentially, MLOps tailored for LLMs provide a framework for managing applications powered by these models [41,42]. The growth of LLMOps highlights the need to refine traditional MLOps to meet the unique challenges of the production scale [43]. Robust LLMOps practices are critical to maintaining performance and reliability throughout the AI lifecycle. The implementation of these practices throughout the lifecycle from data collection and model training to deployment and maintenance is essential to mitigate these risks. Continuous monitoring and proactive detection of anomalies and adversarial behaviors are necessary to secure the integrity of LLMs and their applications [44].

An extensive study has been conducted on MLOps platforms covering principles, architectures, workflows, and challenges. Kreuzberger et al. [18] provided comprehensive analyses of MLOps principles, components, and workflows. Najafabadi et al. [45] reviewed the state of the art in MLOps. However, Faubel et al. [46] presented Industry 4.0 case studies and practical insights. Laurea et al. [47] compared MLOps with DevOps by focusing on open-source tools. Symeonidis et al. [19] explored the integration of MLOps with AutoML. Xu et al. [48] examined the deployments of the ML model (YOLOv3 and LSTM) in AWS and GCP. Eken et al. [49] analyzed 150 academic sources and 48 gray literature sources to identify MLOp challenges, practices, and solutions. Testi et al. [50] emphasized standardized strategies to bridge research and business applications. Atish et al. [51] highlighted the role of CI/CD pipelines in large-scale MLOps deployments. Matsui et al. [52] provided foundational resources for MLOps practitioners. Ayesha et al. [53] discussed the methodologies, benefits, and challenges of MLOps with a practical TensorFlow 2 case study. However, Wazir et al. [54] identified optimal tool structures to implement effective MLOps methodologies.

Figure 1 describes AI types and their associated operations based on degree of specialization.

AIOps focus on the use of artificial intelligence (AI) to automate IT operations tasks to improve system performance and optimize workflows (autoscaling, identifying anomalies, etc.); however, MLOps provide a structured approach to the full lifecycle of machine learning models from training to monitoring in production (model management, model deployment, continuous monitoring, and retraining) [55]. LLMOps integrate GenAI, LLMs, and retrieval-augmented generation (RAG) for efficient development and deployment of advanced LLMs. RAG is a key component in many LLMOps pipelines to enhance the model's ability to generate accurate data. This includes systems for retrieving specific data through vector databases, as well as model fine-tuning, monitoring, inference, and prompt engineering [56]. LLMOps that involve the life cycle management of LLMs include the aspects of GenAIOps (managing model training, deployment, etc.) and RAGOps (managing relevant information to augment generative capabilities) [57]. Figure 1 simplifies overlaps and dependencies between frameworks, and it misses nuances like LLM-specific challenges (scale, ethics, infrastructure, etc.) and RAG's emerging role in improving LLM performance. LLMOps advancements are focused on RAG systems, vector databases, and tools like LangChain for dynamic prompts and context augmentation. The figure misses the details about the role of federated and real-time learning that emphasize adaptability and scalability. All in all, the industry currently prioritizes MLOps but ignores LLMOps. There are still gaps in addressing LLM challenges. In this paper, we leverage LLMOps to provide efficient solutions for managing and deploying LLMs. This research aims to explore the following aspects:

- What platforms and systems can better support LLMs by building on previous MLOps advances?
- Why is it important to address LLMOps challenges not fully managed by traditional MLOps techniques?
- How do LLMOps improve the accuracy of LLMs?
- Why do traditional ML metrics not fully capture LLM performance?
- Why is it important to use LLMOps to improve the performance and accuracy of LLMs?
- How do MLOps and LLMOps differ in terms of their roles in machine learning engineering, and why is it important to understand these differences when addressing challenges in production and deployment? To this aim, key aspects like data management, model development, infrastructure, deployment, system integration, updates and maintenance, versioning, and parallel processing are discussed.

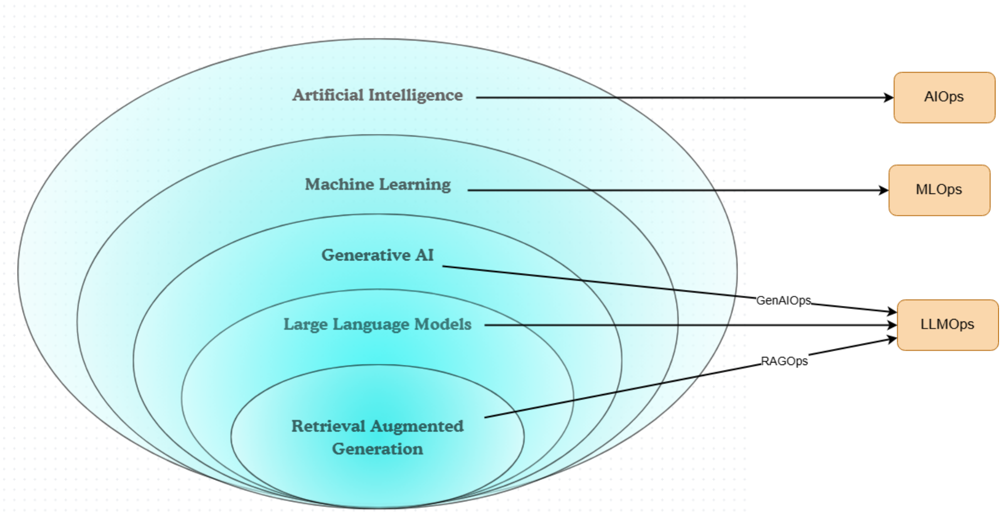

Figure 1. A hierarchy of AI types and associated Ops organized by the level of specialization [57].

Some key contributions of this article in the field of LLMOps are as follows. First, it provides a comprehensive comparison between MLOps and LLMOps. It highlights the distinctive challenges posed by LLMs like scalability, security, and ethical considerations. Second, it evaluates current platforms and tools for LLMOps to offer actionable insights for practitioners. Third, by exploring emerging trends and platforms (e.g., LangChain and Vertex AI), it discusses and offers actionable insights and implications into optimizing LLMs' deployment and performance in different industries (production environments) like healthcare, finance, and cybersecurity. Finally, it outlines future research directions to further refine LLMOps practices and address biases and ethical concerns, as well as improve the security and scalability of LLMs systems.

The remainder of this article is structured as follows. Section 2 presents MLOps. In Section 3, LLMOps are discussed. Section 4 presents DevOps. Section 5 outlines the differences between LLMOps, MLOps, and DevOps. However, Section 6 presents the open issues and future direction. The conclusions are described in Section 7.

2. Machine Learning Operations (MLOps)

MLOps, short for Machine Learning Operations, encompass the procedures and techniques used to install, monitor, and manage machine learning models in production environments with maximum efficiency and effectiveness. This ensures the efficacy, efficiency, and scalability of the models, enabling their application in a productive, cost-effective, and timely manner. The core idea behind machine learning on production (MLOps) is to take a machine learning model that you develop on your own computer and move it into a production environment where thousands of people can use it [58].

2.1. Why Do We Need MLOps?

According to Databricks [59], the deployment of machine learning models in production is difficult. Complex components of the machine learning lifecycle include data import, data prep, model training, tweaking, deployment, monitoring, explainability, and more process involvement, which are a challenge to deploy in production successfully. MLOps require seamless cross-team collaboration, especially between the Data Engineering, Data Science, and Machine Learning Engineering teams [60]. Effectively managing these interdependent processes requires strict operational discipline to ensure continuous and simultaneous execution. As a result, the process encompasses the entire machine learning lifecycle by emphasizing rigorous experimentation, iterative development, and ongoing model optimization [61].

As shown in Figure 2. MLOps have grown into an independent ML lifecycle management strategy, which refers to the end-to-end process of managing ML models from initial phases such as the development phase through the end phase of deployment in the production instance and ongoing maintenance, as well as enhancement in ML models [62]. Raw data are acquired from many sources to start the MLOps lifecycle. The acquiring of raw data is followed by a data analysis that identifies patterns. The data are then cleaned, and features are engineered and versioned for model training. Various algorithms are applied to create and evaluate predictive models. After validation, the model is deployed in production for real-time predictions. Continuous monitoring is needed to ensure performance and maintain accuracy. Also, hardware resources may be scaled to meet demand. Finally, the model is retrained with new data to maintain accuracy, completing the lifecycle [63–65].

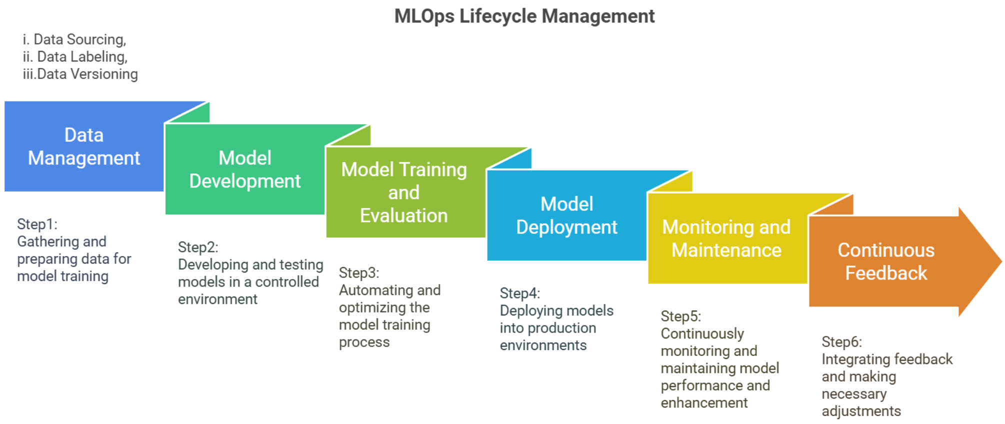

Figure 2. MLOps lifecycle management [66].

Data collection, model generation (within the software development lifecycle (SDLC), including continuous integration/continuous delivery), automation, deployment, health monitoring, diagnostics, governance, and business metrics all come together to create a robust framework in MLOps [66]. Building this pipeline that addresses the five key performance diagnostic challenges (i.e., data governance, model deployment, model training, sensitivity to alarm thresholds, and explainability) helps to effectively overcome these challenges [65]. Figure 2 does not show LLM challenges like prompt engineering, fine-tuning, and ethics, as well as the iterative and dynamic nature of model development that requires continuous learning and adaptation. Recent research highlights the importance of Human-in-the-Loop (HITL) systems and CI/CD pipelines in LLMOps. These improve accuracy and reduce biases. The use of AutoML and Model Monitoring ensures scalability. MLOps should expand to include LLM-specific stages like prompt optimization and ethical auditing.

2.2. Benefits of MLOps

Yasir et al. [67] identified 58 distinct Critical Success Factors (CSFs) related to MLOps projects that were categorized into three key areas: technical, organizational, and social/cultural dimensions. Tazeem et al. [62] emphasized the importance of strong MLOps plans to guarantee effective security standards for ML deployments in operational technology settings. Joshi [68] presented case studies that demonstrate successful deployments in various industries and demonstrated the tangible benefits of this approach in real-world applications. Jana et al. [51] emphasized the critical role of version control and repeatability in ensuring the traceability and longevity of deployed ML models. These framework simplify the deployment of ML models in a reliable, scalable, and efficient manner. They improve model quality, increase automation, and foster better collaboration between data scientists, developers, and operations teams, as well as improve monitoring and maintenance and reduce hazards related to machine learning in production environments. The key benefits are described below.

2.2.1. Automation and Scalability

MLOps form a field that automates the entire machine learning model lifetime, cutting down on human intervention. Automation improves efficiency and dependability in machine learning applications by managing the entire model lifecycle, ranging from training, testing, deployment, and monitoring [63,69]. In fact, MLOps frameworks such as MLflow, Kubeflow, and Airflow are specifically designed to handle large-scale model deployments with ease and scalability. These frameworks enable efficient management of models across diverse contexts and systems, thereby playing a pivotal role in ensuring operational effectiveness in large-scale deployments [51].

2.2.2. Continuous Integration and Deployment (CI/CD)

By incorporating CI/CD ideas into machine learning, MLOps enable continuous model deployment, integration, and testing. That way, whenever new data are generated or changes happen, models can be easily retrained and redeployed because they are constantly up to date. By guaranteeing the safe release of updated models into production systems through automated testing and validation, continuous deployment reduces downtime [60,63,70].

2.2.3. Monitoring and Performance Tracking

Monitoring and performance evaluation are critical components of MLOps, as they ensure that machine learning models operate effectively in production and minimize potential business or customer impacts. Key practices include real-time monitoring, data drift detection, alerts, and notifications, as well as comprehensive logs and auditing [49,63].

2.2.4. Improved Model Quality and Reduced Risk

Models are protected from hazards like data drift and security vulnerabilities by continuous testing, validation, and monitoring of MLOps. To reduce the chances that biased models are used in production settings, automated pipelines can have built-in tests for model correctness and fairness [60,62,69].

2.2.5. Improved Collaboration Between Teams

By leveraging MLOps standardized workflows and communication channels, data scientists, ML engineers, and operations teams are able to work together more effectively. The pressure on the teams responsible for development and deployment is reduced as a result, and there are smooth transitions between model development, testing, deployment, and maintenance. This integrated approach improves coordination between various teams and improves the scalability of the system in production [71–73].

2.2.6. Compliance and Governance

MLOps frameworks typically provide functionalities that facilitate regulatory compliance and governance over data utilization, model performance, and deployment procedures. To help with regulatory compliance, foster stakeholder collaboration and co-learning, as well as ensure the safe implementation of novel public sector AI services, the methodology proposes the use of AI regulatory sandboxes and Machine Learning Operations practices [74].

2.2.7. Data Management and Versioning

In MLOps, the data are versioned, along with the models and code, which guarantees that any model can be retrained or validated using the precise version of the data with which it was originally developed. This guarantees data integrity and traceability, which are essential for debugging and auditing the performance of the model over time [75–77].

2.2.8. Simplifies Complex ML Workflows

MLOps mitigate the intricate nature of machine learning workflows by offering a systematic framework to manage various processes, including feature engineering, data pretreatment, model training, and deployment. This facilitates the management of dependencies across several contexts, ensuring uniformity in workflows from development to production. The pipelines enhance the end-to-end life cycle of machine learning models [78,79].

2.3. Applications of MLOps

For effective deployment, maintenance of models in production, and integration into real-world systems, the MLOps framework is vital for operationalizing machine learning models. Many different types of business can simplify their processes, boost their efficiency and effectiveness of ML model, and improve the performance and computing load. It also streamlines complex ML workflow, and deployment challenges issues are simplifying by using this approach. Presented below are potential applications of this methodology:

- Provide effective security standards for ML implementations in complex operational technologies using this methodology [62].
- Using the help of MLOps technologies, industrial settings can improve their image recognition accuracy and adapt well to new conditions [80].
- Automating model training and deployment, as well as integrating these processes into typical CI/CD pipelines, which is crucial to address the challenges associated with the effective deployment of machine learning models using the MLOps methodology [63].
- The MLOps principles are particularly advantageous for large projects that require continuous deployment and robust automated operations [81].
- Similar approaches can be used by MLOps cross-domain applications in healthcare and finance to effectively manage changing data streams and concept drift [82].
- This study presents a resilience-aware MLOps approach for AI-powered healthcare diagnostic tools. Its primary goal is to make systems more resistant to harmful outside forces, such as hostile attacks and drift [83].
- It could be used in various applications, particularly by using microscopic pictures. For example, the study in [84] investigated the use of MLOps analysis of sparse image data and introduced a comprehensive approach that employs fingerprinting to select optimal models and datasets. The method also employs automated model development while leveraging continuous deployment and monitoring to facilitate learning from errors.
- The use of MLOps for the prediction of lifestyle-related diseases. Through the analysis of massive volumes of diverse healthcare data, this helps to predict lifestyle diseases, which in turn helps to plan prevention, diagnosis, and treatment [85].
- It addresses the challenges of model retraining and versioning, as well as ensures that the model remains efficient and more effective over time, resulting in the rise of MLOps integrating to everyday applications such as smart kitchens and radiology systems to detect turbine performance. These functions mitigate operational challenges, collaboration challenges, and deployment challenges to build an intelligent application using this methodology [86].

Table 1 outlines various existing MLOps platforms, highlighting the key features, and providing examples of the use case to give a deeper understanding of the landscape.

Table 1. Comparison of MLOps platforms for different use cases.

Platforms	Key Features	Use Case	Focus	References
AWS Sagemaker (Python SDK version 2.94.0)	Fully managed infrastructure, workflows, and tools for building, training, and deploying machine learning (ML) models for any use case.	Suitable for enterprise-level ML applications and workflows.	Scalability, Integration with AWS ecosystem (S3, EC2, and other services).	[87–94]
Databricks (Version 10.4)	A comprehensive analytics platform that offers collaborative, real-time notebooks, scalable data processing, and integrated machine learning workflows.	Well-suited for the analysis of large-scale datasets (Structured, Unstructured) and the collaborative construction of machine learning models (Building, Training, and Deployment).	Integration with Apache Spark, Collaborative notebooks supports multiple languages (Python, R, Scala, SQL).	[95–99]
Azure AutoML (Azure ML SDK version 1.38.0)	Automated model building and tuning, support for many machine learning tasks, and seamless interaction with Azure services; cost effective.	Ideal for individuals who need to quickly create, train, and use machine learning models, as well as time series forcasting.	Support for a variety of data types and models, integration with Azure ML, time series specialization, automation and accessibility.	[100]
TensorFlow (Version 2.10.0)	Comprehensive machine learning framework, distributed training with (TensorFlow Distributed Strategy), model serving (TensorFlow Serving) and edge deployment with (TensorFlow Lite).	Machine learning development from beginning to end, scalable training, production deployment, mobile and edge inference technology.	Management of machine learning pipelines, end-to-end development and training of machine learning models, serving models at scale, enables fast training and cross platform compatibility (mobile devices, cloud environment, etc.).	[101]
PyTorch (Version 1.13.0)	High-level application programming interfaces (TorchVision, TorchText, etc.), model serving (TorchServe), combination with Kubernetes and cloud platforms.	The use of research and experimentation, the construction of flexible models, and production deployment. Provides dynamic methodology for constructing and training neural networks.	Deployment of models at scale, efficient deployment, scalable training (distributed data parallel).	[102–104]
MLFlow (Version 2.0.0)	MLflow is a popular MLOps platform with full machine learning model lifecycle management tools. It simplifies ML project management with experiment tracking, model versioning, and reproducibility.	MLflow is adaptable, addressing various machine learning scenarios in model selection, deployment, model performance monitoring, model versioning, and management.	Cross-Platform Integration: MLflow interacts with various machine learning frameworks, Scalability and Flexibility, Cloud-native.	[105–108]
Kubeflow (Version 1.4.0)	Open-source MLOps platform Kubeflow runs scalable and portable machine learning workloads on Kubernetes. It manages model construction, training, deployment, and monitoring for ML.	Kubeflow facilitates the automation and scaling of recommendation models for e-commerce, media platforms, fraud detection models, and bioinformatics application utilizing Kubeflow.	Cloud-Native and Kubernetes Integration, Scalabilty, Automation, and Reproducibility.	[105,107,109–111]

Table 1. *Cont.*

Platforms	Key Features	Use Case	Focus	References
Metaflow (Version 2.5.0)	Human-centric MLOps platform Metaflow simplifies data science project development and management. Focus on code while handling scaling, versioning, monitoring, and simplifying model development.	Metaflow is a user-friendly library that assists scientists and engineers in constructing and overseeing practical data science projects. Netflix is capable of holding numerous Metaflow projects.	Diverse integrations for Metaflow, Parallelization and Resource Optimizations, Collaboration, and Transparency.	[112–114]
IBM Watson Studio (Version 3.0.1)	IBM Watson Studio enables analysts, data scientists, and developers to collaboratively build, train, and deploy machine learning models.	IBM Watson Studio is a comprehensive platform that merges many tools and technologies to streamline the development, deployment, and maintenance of machine learning models, and it facilitates with NLP and AI projects.	IBM Watson Cloud services are used to develop natural language processing solutions, and it enables the creation of advanced chat tools, as well as support for open-source tools.	[115,116]
Cloudera (Version 7.1.4)	Cloudera manages machine learning models in production at scale, model monitoring, ETL capabilities, and governance tools.	Comprehensive control of the entire machine learning lifecycle; facilitates visibility throughout the whole machine learning lifecycle.	Cloudera provides open standards-based MLOps enabling enterprises to industrialize AI, big data processing and analytics solutions, and enterprise data management.	[117,118]
Apache Airflow (Version 2.3.0)	Apache Airflow is a widely utilized tool for orchestrating intricate, multistage data pipelines and workflows across many sectors, especially in data engineering, machine learning, and ETL operations.	Apache Airflow has been utilized for the orchestration of ML operations and the scheduling of automated model training, as well as batch processing.	Automation of machine learning pipelines for anomaly detection challenges, data migration and integration, and task scheduling and monitoring.	[105,119–122]

2.4. Challenges in MLOps

Implementing MLOps involves various challenges that range from technical complexities to organizational and operational hurdles. For example, Faubel et al. [46] presented a case study in Industry 4.0, as there is limited information on its practical deployment in industrial businesses. While only a small percentage of ML models make it to production, an investigation conducted by Databricks highlighted the challenges associated with difficult handoffs, security and compliance risks, and complexity regarding managing ML environments [59].

The lack of standardization in ML tools and procedures limits the transition from model development to production, highlighting the importance of resilience features such as transparency and security [123]. Data preparation, experimentation, and continuous monitoring create a challenging workflow for machine learning engineers (MLEs), which can overload teams and cause deployment failures. It requires strong data science engineering skills to assist with ML engineers [124,125]. Future research should focus on developing the MLOps field by bridging the gap between business objectives and modeling perspective through appropriate frameworks, as the success of data science projects depends not only on technical matters [126]. Adopting MLOps presents many hurdles for practitioners, including the complexity of ML solutions, platform challenges, pipeline complexities, and organizational diversity. Addressing these challenges is crucial for understanding the adoption barriers of these methodologies and developing effective solutions [49]. The primary challenges of MLOps adoption in enterprises are collaboration between ML, DevOps, operations (Ops), science teams, and data teams. Conceptual changes are obstacles to workflow automation, and employers often lack a complete understanding of the paradigm [109].

3. Large Language Model Ops (LLMOps)

3.1. What Are LLMOps?

LLMOps (Large Language Model Operations) involve an innovative methodology designed to tackle the issues of implementing LLMs in practical applications [127]. It is essential to improve the precision of the recommendations, minimize latency, and improve user experiences. LLMOps guarantee the efficient functioning of LLMs in recommendation systems by fine-tuning the model, enhancing user experience and improving computational processes [43]. Today, future trends are building on revolutionizing the AI industry with LLMs, and a different approach is required to maintain AI-powered products. As a result, new guidelines and resources will emerge to manage the lifecycle of LLM-driven applications [128]. These approaches facilitate the seamless deployment, monitoring, and retraining of LLMs for organizations, offering a complete framework and best practices for AI practitioners seeking to efficiently operationalize their generative AI systems [129]. This methodology intended for the management and maintenance of ultra-large-scale machine learning technology. It enables the automated deployment and management of machine learning models through natural language, improving efficiency and dependability [43].

3.2. LLMOps Life Cycle Components

Based on model selection, model tuning, deployment, prompt engineering, and monitoring, the fundamental components of LLMOps are crucial to efficiently oversee the lifetime of LLMs across diverse applications. These components are engineered to address the distinct issues presented by LLMs, including scalability, security, and ethical considerations while improving their operational efficiency and dependability [127,130]. The main elements of LLMOps, as defined in the article [41], are organized around the Discover, Distill, Deploy, and Deliver (4D) phases within its framework. These elements are crucial for efficiently overseeing the lifecycle of LLMs in enterprise environments. This GenAI solution focused on the creation and deployment of LLMs-based applications. LLMOps promise to create strong, high-performance LLM systems that can oversee comprehensive operations, including managing vector databases. The Figure 3 illustrates the LLMOps architecture, which comprises multiple key components designed to enhance the interaction between users and LLMs using RAG. The process begins with user-submitted queries (prompts) that are inputs formulated by users to request specific information or perform tasks. These prompts are transferred to the embedding model. It converts the text into a dense numerical representation (i.e., embeddings) to capture the query's semantic meaning. Then, the system uses RAG mechanisms to search a vector database using the embeddings generated from the query. The vector database stores a vast array of preprocessed information. The various sources of information include structured documents, unstructured data, and multimedia content, which are indexed in a format optimized for efficient similarity matching. The RAG procedure retrieves the most relevant information from the database. The retrieved information is appended to the initial user query via a procedure called context augmentation. It enriches the input with additional data to provide a more comprehensive understanding for the model. Using this augmented context, the LLMs processes the integrated data and generates the final output. This architecture ensures the accuracy of responses and enhances the overall user experience by utilizing the capabilities of LLMs and RAG systems. Figure 3 highlights RAG's role in improving LLM outputs but omits complexities such as training challenges, fine-tuning, and distributed training in LLMOps. Recent LLMOps advancements include distributed training (e.g., DeepSpeed, ZeRO, etc.) model parallelism, and ethical AI frameworks for security and fairness. It is thus necessary to update the architecture with ethical auditing, adversarial testing, and distributed infrastructure.

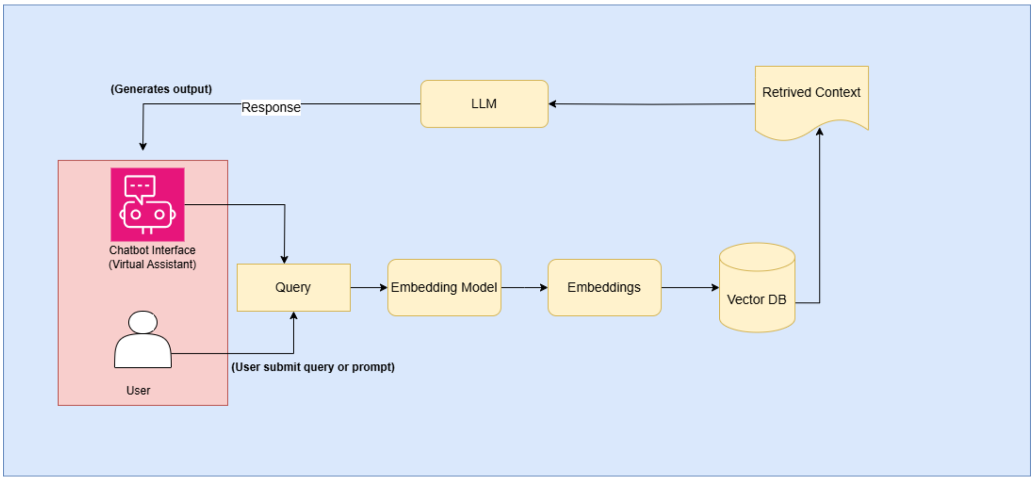

Figure 3. LLMOps architecture pattern: enhancing virtual assistant interactions with RAG [131].

3.3. Why Do We Need LLMOps?

Early LLMs like BERT and GPT-2 were introduced in 2018, showcasing the impact of transformer-based architecture, but initially did not gain popularity due to the narrower application scope. As shown in Figure 4, the demand of ChatGPT (with Model version GPT-3.5) in December 2022 led to a surge in media attention, and LLMs were subsequently integrated into a broad range of applications in various domains. These include content generation (e.g., ChatGPT [132]), program assistance (e.g., GitHub Copilot [133,134]), writing assistant (e.g., Notion AI [135] (various model versions are available Notion 2.21, Notion 2.47), Jasper [136] (multiple integration patterns, AI application library available to automate various generative tasks)) and other areas that have revolutionized the AI industry (e.g., LLaMA and Gemini) [137,138]. By 2023, various industries, including blockchain, security, and data utilization, began to embrace LLMs. This change shows how media publicity can significantly influence their adoption across in different industries, extending their use from chatbots to fields such as finance, data technology, and semiconductors [139]. Figure 4 did not cover the challenges (e.g., costs, ethics, and model drift) or multimodal LLMs combining text, vision, and audio.

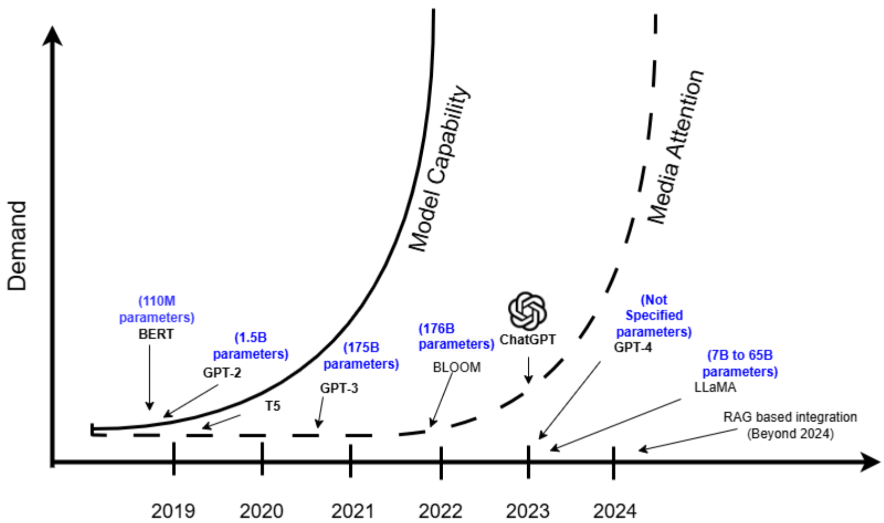

Figure 4. Rise of LLMs [128].

People are sharing their experiences as they develop and deliver LLM-powered applications to production. Although creating something innovative with LLMs is relatively simple, preparing it for production presents a significant challenge [140]. It shows the importance of combining DevSecOps with LLMOps, ensuring that security is a shared duty throughout both the development and the operational phases. This provides both theoretical and practical support for the successful implementation of LLMs [130]. Using LLMOps, enterprises can improve the efficacy and reliability of large-scale machine learning models, resulting in personalized recommendations that are more closely aligned with user preferences [43]. As depicted in Figure 5, the power of LLMOps lies in its ability to effectively manage and optimize the LLM lifecycle in the production environment [43,127,141–143].

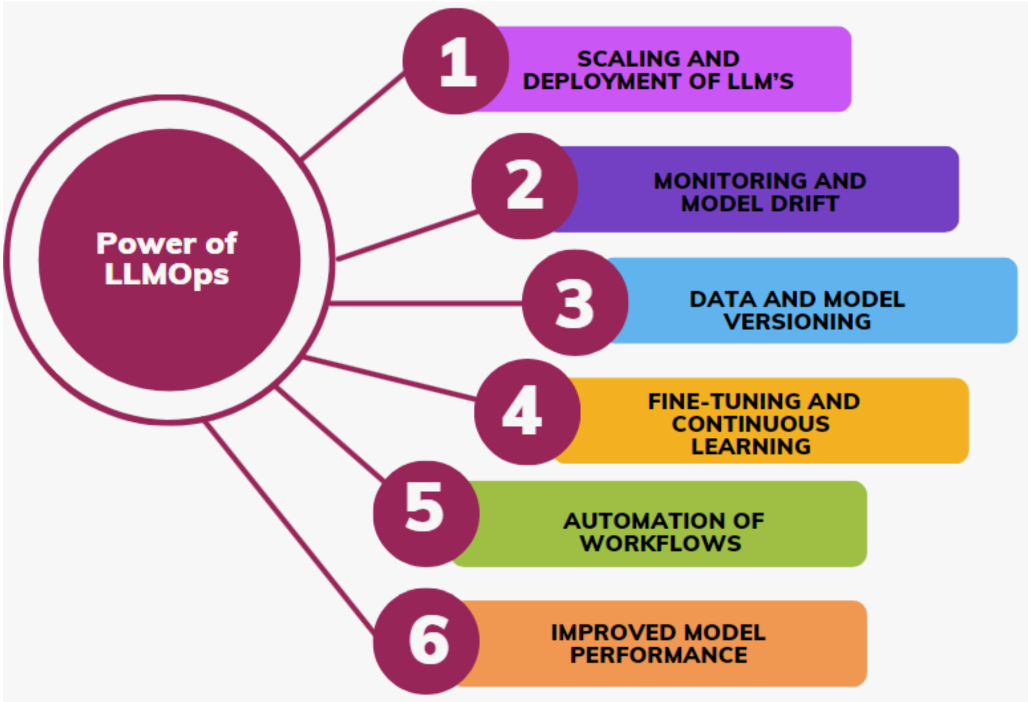

Figure 5. Power of LLMOps.

Implementing LLMOps offers several advantages to organizations that use LLMs, significantly influencing their AI efforts and overall commercial results. Implementation enhances team productivity through various steps, beginning with collaboration among team members. Data scientists, ML engineers, DevOps, and stakeholders can engage more efficiently on a consolidated platform for communication and insight exchange, model development, and deployment, leading to faster delivery [43]. LLMOps constitute a specialized component of FMOps (Foundation Model Operations) that enhances the principles of MLOps (Machine Learning Operations) to facilitate the seamless deployment, monitoring, and retraining of LLMs within enterprises [129]. The LLMOps framework, which is organized around the Discover, Distill, Deploy, and Deliver (4D) stages, offers a systematic approach to the management of the LLMs lifecycle, which enhances operational reliability [41]. The study concludes that the LlamaDuo pipeline represents an important breakthrough in LLMOps, offering a robust framework for transitioning from large-service LLMs to smaller ones, thus ensuring service continuity in operational failures, rigid privacy policies, or offline requirements [141].

3.4. Best Practices for LLMOps

As can be seen in Figure 6, there are ten key areas of best practices for LLMOps. These include data management that ensures clean and high quality data for training, as well as model training and fine-tuning that involves continuous improvement to LLMs for specific use cases. For Scalability and Infrastructure, scalability highlights the need for robust systems that handle growing demands and robust infrastructure for efficient deployment and management. Monitoring and Observability ensure that model performance and behavior are tracked. Security and Compliance focuses on protecting data and adhering to regulations, and Inference Optimization addresses the efficiency of LLMs deployment. Continuous development is supported by Continuous Integration/Continuous Deployment (CI/CD) to ensure seamless updates. Collaboration and documentation promote team alignment. Human-in-the-Loop (HITL) integrates human oversight for quality control. Ethical considerations emphasize responsible AI use by addressing biases and ensuring fairness. All in all, Figure 6 did not include direct details about dynamic development needs, prompt engineering, and adversarial robustness, which are key for performance and security. Recent LLMOps progress emphasize HITL systems, adversarial testing, and CI/CD pipelines to enhance LLM accuracy, reduce biases, and ensure scalable deployment.

Figure 6. Best practices for LLMOps [144].

3.5. Applications of LLMOps

LLMOps streamline repetitive, labor-intensive activities, facilitating expedited processing across several domains (automation process). They facilitate the extensive deployment of LLMs in practical applications, guaranteeing that models can manage substantial volumes of data and user interactions (scalability). Additionally, they aid in facilitating the development and administration of intelligent systems capable of processing and generating natural language, hence enhancing decision making in sectors such as finance, healthcare, and cybersecurity (intelligent decision making). Figure 7 shows a list of applications used in LLMOps is provided in various sectors. Its features are described as follows:

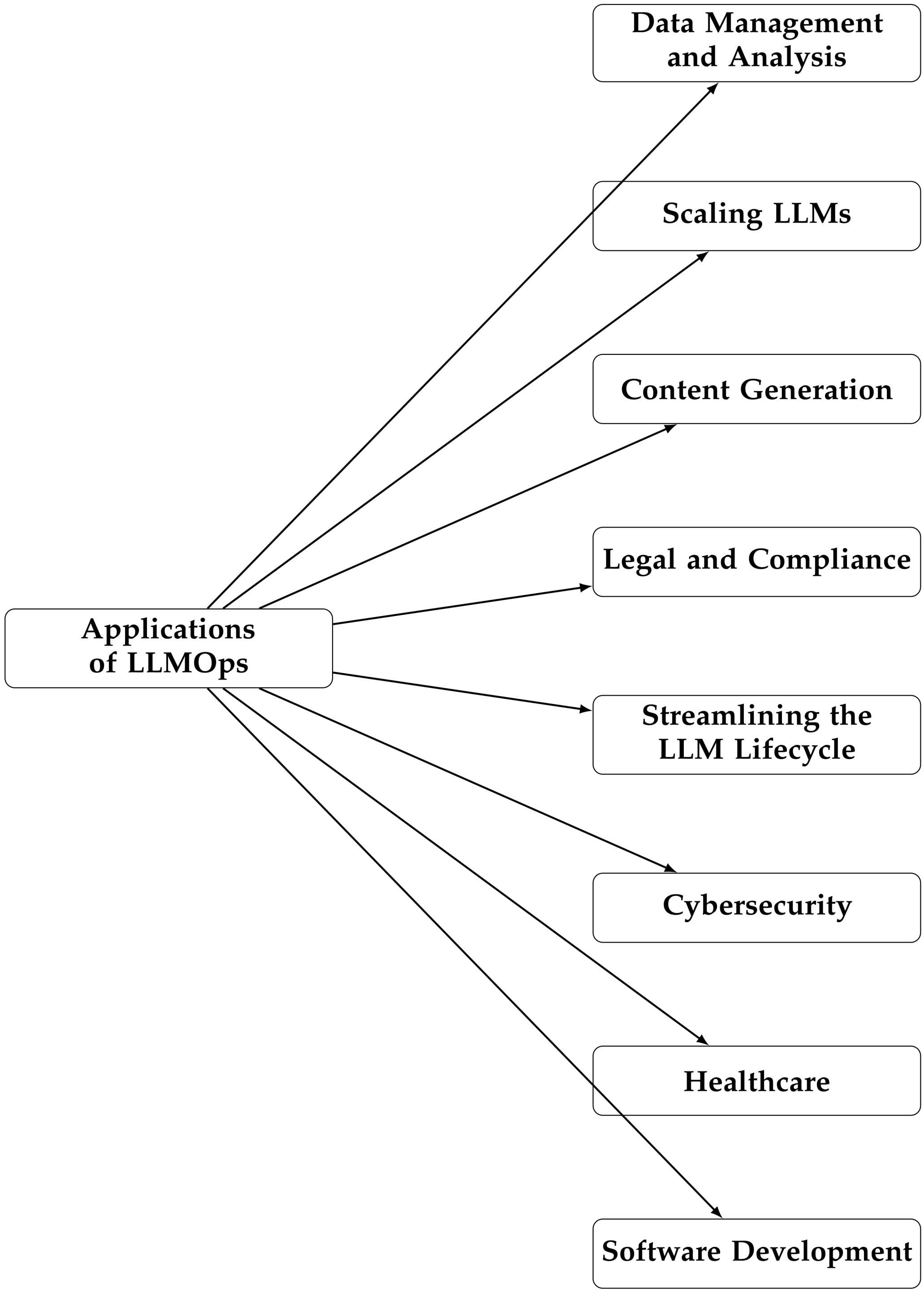

Figure 7. Tree chart of LLMOps applications [130,145–164].

- The document [165] presents the concepts of LLM–Computer Interaction (LLMCI), in which LLMs integrate with computer vision to engage with user interfaces. The applications of LLMOps encompass the facilitation of LLMs in comprehending and manipulating UI elements, retrieving information, executing functions, and doing duties analogous to human interactions. LLMOps enable more human-like interactions with computers by integrating language comprehension and visual perception capabilities.
- The study in [166] presents a framework for LLMOps, which is a distinct subset of MLOps targeted to the development, deployment, and maintenance of LLMs within

Continuous Integration/Continuous Deployment (CI/CD) pipelines. The case study on text summarization demonstrates that integrating a human feedback loop into the LLMs CI/CD pipeline improved its quality and dependability, emphasizing the importance of human input in improving LLMs performance.

- The work in [43] enhances the user experience in personalized recommendation systems by refining extensive machine learning models to provide accurate, timely, relevant, and precise recommendations based on the interests of each individual user, as well as integrating prompt engineering. LLMOps customize input prompts to improve recommendation accuracy and user happiness.
- The work in [167] indicates a reference framework for the development of a large language models (LLMs) application stack, highlighting common systems, tools, and design methodologies identified in companies and AI startups.

3.6. LLMOps Platforms for Managing Large Language Models

The landscape of LLMOps is a rapidly growing ecosystem that combines various platforms and frameworks to manage the lifecycle of LLMs. The key components of this landscape include platforms for model deployment and scaling, frameworks for model customization and integration, tools for monitoring and ethical considerations, and automation frameworks for continuous integration and deployment. As LLM technology continues to advance, the landscape will evolve, with emerging trends such as federated learning, model distillation, and real-time learning contributing to the optimization of deployment at scale. Table 2 outlines various platforms, highlighting their key features and providing examples of the use case to provide a deeper understanding of the different platforms in the LLMOps landscape.

Table 2. Comparison of LLMOps platforms for different use cases.

Platforms	Key Features	Use Case	Focus	References
Hugging Face Transformers (Version 4.30.0)	It comes with a model hub, API support, and pretrained models for a number of NLP tasks. It is also easy to integrate with PyTorch and TensorFlow.	This is the best way to quickly build and change natural language processing (NLP) models, multimodal applications, and text summarization.	Fine tunning of NLP uses, model modification, scalability, and performance.	[130,168–170]
LangChain (Version 0.0.150)	Multiple LLMs can be used together, including OpenAI, Hugging Face, and others. Interactive agent and prompt management are supported.	Perfect for making complicated LLM apps with dynamic prompt management and processes based on agents, dynamic content generation.	Automation of complex workflows, the creation of chatbots, and managing context for LLMs; natural language understanding, text generation, and text classfications.	[171–178]
DeepSpeed (Version 0.8.0)	ZeRO optimization, mixed precision training, gradient accumulation, and distributed training are some of the features that are included in optimized training for big models.	Ideally suited for the efficient and effective training of large-scale deep learning models with limited resources.	Suitable for the training of large-scale deep learning models, scalability for large models, simplified distributed training.	[179–181]
Google Cloud Dialogflow CX (Version 2.0.1)	Visual flow builder, multiturn discussions, enhanced natural language understanding, support for omnichannel use, context management.	Ideally suited for the development of complicated conversational agents and chatbots that can handle numerous turns across multiple platforms.	Virtual assistants, customer care bots (optimize the customer experience), seamless integration with Google Cloud platform.	[182–185]

Table 2. *Cont.*

Platforms	Key Features	Use Case	Focus	References
Azure Bot Service with OpenAI (Version 1.0.0)	Integrated bot framework, support for OpenAI models (such as GPT series), powerful artificial intelligence capabilities, bot orchestration.	Outstanding for the development, deployment, and management of intelligent conversational agents through the utilization of OpenAI's large language models (LLMs).	Automating and scaling customer interactions, built-in Azure services for scalability, integration with a variety of communication channels.	[186,187]
Apache Airflow (with LLMOps) (Version 2.3.0)	Orchestration of workflows, dynamic scheduling of tasks, interaction with machine learning pipelines, and supports distributed execution.	A perfect solution for the management and automation of large language model (LLM) pipelines that contain complicated dependencies, serving and deploying LLMs.	Automation of complex workflows, integration with various ML tools, scalability for large scalable models.	[188,189]
AWS Inferentia (Neuron SDK version 1.7.0)	Designed for deep learning models, high throughput, low latency, cost-efficiency, and multimodel support.	Designed to facilitate the deployment of LLMs and other deep learning models at reduced costs while maintaining high-performance inference efficiency and speech recognition assistance.	Providing real-time inference for LLMs, scaling at a cost-effective rate, and accelerating deep learning models; integration with AWS ecosystem.	[190,191]
Google Cloud Vertex AI (Version 1.12.1)	Machine learning platform that covers the entire process, including AutoML, managed machine learning pipelines, individualized model training, model monitoring, and tracking.	This solution is perfect for constructing, training, deploying, and administering machine learning models and LLMs at a distributed scale, as well as NLP tasks (text classifications, chatbots, etc.).	End-to-end machine learning platforms, scalability and flexibility, model management and monitoring.	[192,193]

4. Development and Operations (DevOps)

What Are DevOps?

DevOps are a combination of Development and Operations, which represents a new approach in the software engineering field [194]. They entail a cultural and collaborative methodology that unifies software development (Dev) and IT operations (Ops) to improve communication, cooperation, and efficiency across the software development lifecycle [195]. This methodology is designed to integrate development and operations teams within organizations, promoting expedited software delivery and improved collaboration [196]. The objective is to improve software delivery performance and improve team engagement while overcoming the weaknesses of conventional software development methodologies [197].

The DevOps lifecycle consists of seven key components (also known as 7 Cs), as illustrated in Figure 8. These include Continuous Development, Continuous Integration, Continuous Testing, Continuous Deployment/Continuous Delivery, Continuous Monitoring, Continuous Feedback, and Continuous Operations. Table 3 outlines the DevOps platforms and frameworks for various use cases. Figure 8 does not explicitly address aspects critical to LLM integration, such as prompt engineering, model fine-tuning, or ethical AI practices in LLMOps. Recent researche stresses LLM-specific CI/CD pipelines with prompt engineering, fine-tuning, and ethical auditing. DevSecOps adoption is key for secure LLM deployment. It suggests that DevOps should include LLM stages such as prompt optimization and ethical checks.

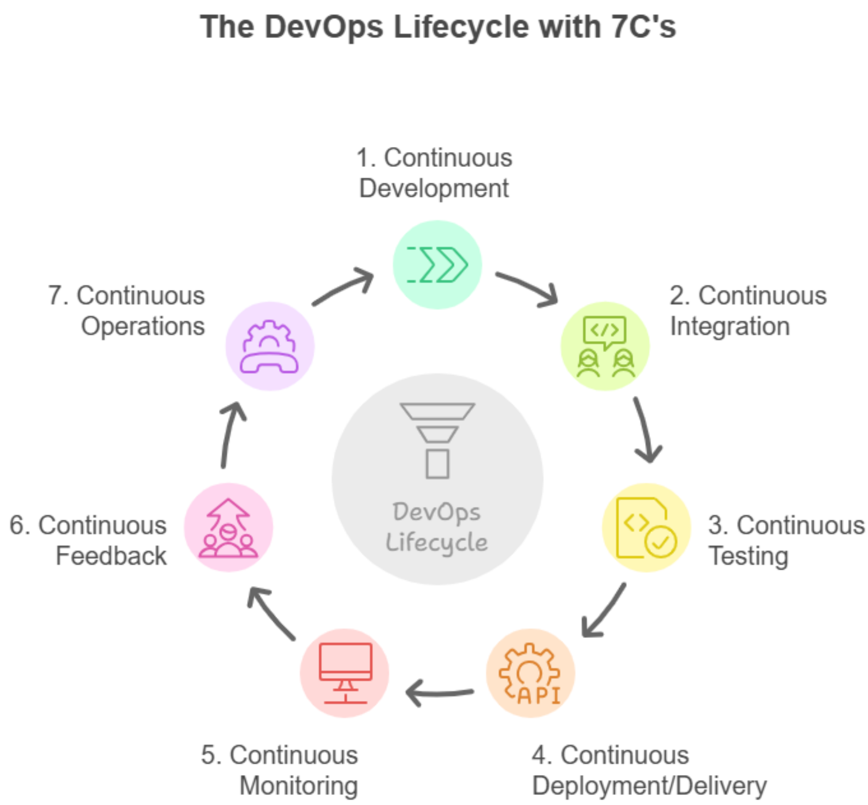

Figure 8. DevOps lifecycle with 7 Cs [198].

Table 3. Comparison of DevOps platforms for different use cases.

Platforms	Key Features	Use Case	Focus	References
Jenkins (Version 2.319.3)	Jenkins is a free and open-source automation tool designed to streamline Continuous Integration and Continuous Delivery (CI/CD) processes, and it features support for development, testing, and deployment.	Automated development and testing, multienvironment deployment, customized workflows for larger enterprises, infrastructure automation.	Flexibility allows users to construct customized pipelines, scalability, and distributing tasks; integration with other DevOps tools.	[199–202]
GitLab CI/CD (Version 15.5.0)	GitLab covers full DevOps lifecycle, including source control to deployment. Complete DevSecOps capabilities.	GitLab manages Continuous Integration (CI) operations, multicloud deployment, and automated testing.	Ideal for enterprises utilizing GitLab for source control and those requiring DevSecOps; flexibility and customization.	[203–205]
CircleCI (Version 2.1.0)	CircleCI is a cloud-based Continuous Integration (CI) service that streamlines software development by automating the build process.	CircleCI can be integrated to Google Firebase Test Lab to test Android apps on multiple devices and configurations using cloud infrastructure.	Effective for small- to mid-sized projects with fast, scalable CI/CD; infrastructure optimization.	[206–208]

Table 3. *Cont.*

Platforms	Key Features	Use Case	Focus	References
Azure DevOps (Version 2.0.0)	Azure DevOps is Azure Pipelines, which was first made on Microsoft Team Foundation Server. It has grown into a strong environment that Microsoft uses a lot for software development.	Azure DevOps has various components, such as Azure Boards, Repos, Pipelines, Artifacts, and Test Plans; makes software creation faster and easier.	Microsoft builds the majority of its software with Azure Pipelines, which allows the company to take advantage of the stability, resilience, agilty, and collaboration.	[209–211]
Kubernetes (Version 1.31.5)	A Kubernetes pipeline indicates the automated processes within a CI/CD framework that utilizes Kubernetes for the deployment, scaling, and management of containerized applications.	It focuses on the development cycle, which includes writing code, compiling it, testing it, and fixing bugs, all while Kubernetes runs apps in containers.	Optimal for the orchestration, scaling, and automation of containers in distributed environments.	[212–215]
Docker (Version 20.10.9)	Docker DevOps pipeline enhances automation and environment replication by integrating containerization into Continuous Integration (CI) workflows.	Utilizing Docker enables teams to establish uniform environments, optimize application delivery, and promote swift iterations.	Ideal for using containers to create, execute, and share isolated application environments; simplifies application deployment.	[216–218]
SonarQube (Version 9.6.1)	SonarQube provides a solution that automates white-box testing and security assessments within a continuous integration (CI) pipeline.	SonarQube enables developers to effectively oversee vendor branches and guarantee that both their proprietary code and third-party components adhere to security and testing best practices.	Optimal for maintaining code quality, identifying defects, and guaranteeing adherence to coding standards.	[219,220]

5. Difference Between LLMOps, MLOps, and DevOps

DataOps offer tools for creating efficient data processing pipelines, commonly known as DevOps. MLOps establish a systematic framework for the building, training, evaluation, optimization, deployment, inference, and monitoring of machine learning models in a production environment. LLMOps serve as a comprehensive framework that integrates components of generative AI (GenAI), large language models (LLMs), and retrieval-augmented eneration (RAG). Figure 9 represents an intersection diagram comparing DevOps, MLOps, and LLMOps. DevOps emphasize software development, operational processes, automation, and Continuous Integration/Continuous Deployment pipelines. In contrast, MLOps encompass the full machine learning lifecycle, comprising model training, deployment, and monitoring. LLMOps, on the other hand, focus on the optimization, inference, and the handling of LLMs. LLMOps advancements include LangChain, Hugging Face Transformers, RAG systems, vector databases, and federated learning, enabling scalability and continuous adaptation.

The intersections of DevOps and MLOps include automation, scalability, resource optimization, and continuous deployment. Both DevOps and LLMOps focus on model deployment, the security of infrastructure, and the orchestration of models and applications. MLOps and LLMOps emphasize the management of large datasets for LLMs, the supervision of large-scale machine learning models, and the fine-tuning and conduct of experiments. All areas (i.e., automation, monitoring, scalability, interdepartmental collaboration, and performance enhancement) are essential in achieving success. In the case of data management and handling large datasets, MLOps include flexible storage, preprocessing, data management strategies, and the ability to handle different amounts and types of data well. Implementing techniques like data labeling, version control, and the use of different paths to efficiently handle data across multiple machine learning models is part of the process in MLOps. On the other hand, LLMOps address the unique challenges of working with very large and complicated datasets, usually made up of text. These datasets need special ways to be prepared, stored, and managed. LLMs are very big and complicated, so they need high-throughput pipelines, distributed storage systems, and powerful data versioning. The main goal is to efficiently handle large amounts of data while keeping the data's quality and value for improving LLMs. Tables 4–6 provide a detailed comparison of LLMOps, MLOps, and DevOps.

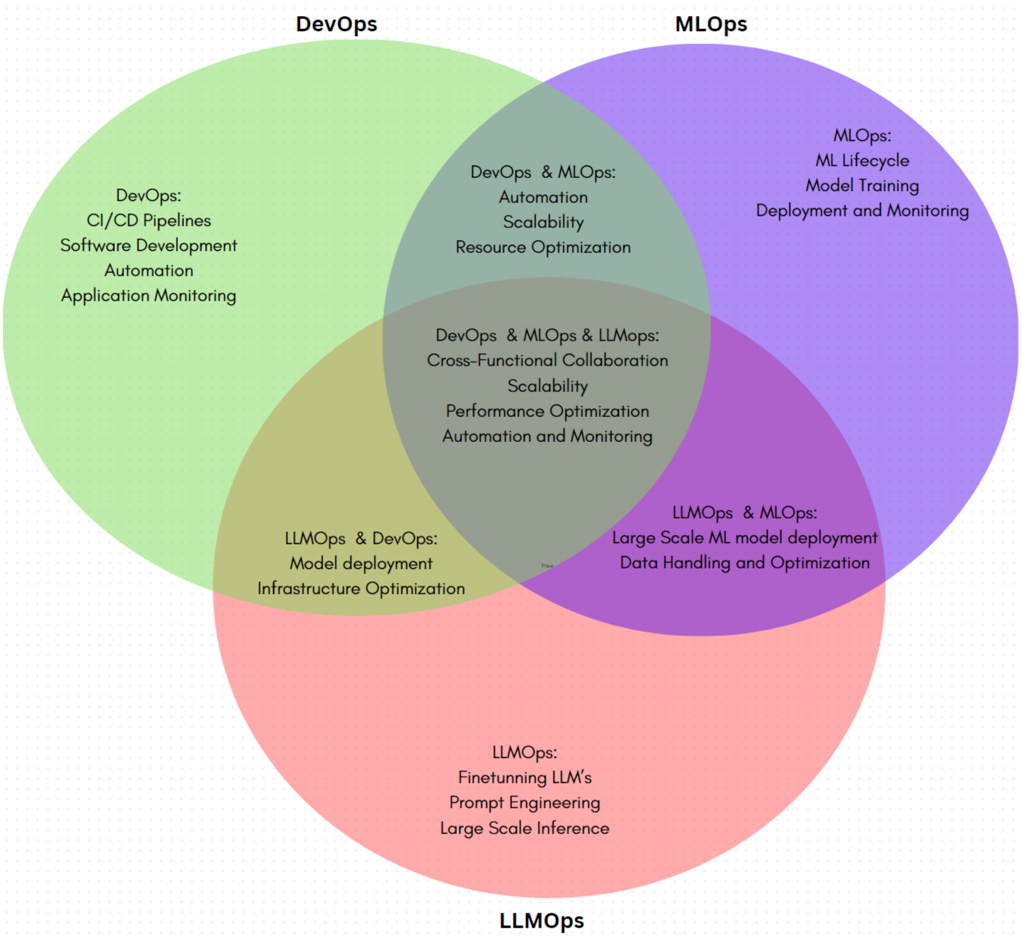

Figure 9. Intersection diagram comparing DevOps, MLOps, and LLMOps [221].

Table 4. Comparison of LLMOps, MLOps, and DevOps.

Aspect	LLMOps	MLOps	DevOps	References
		Hardware Requirements		
GPUs (Graphics Processing Units)	Due to the size and complexity of LLMs like GPT-4, GPUs are essential for LLMOps performance.	GPUs are commonly utilized in MLOps to accelerate deep learning model training. MLOps workflows can scale across GPUs, depending on model size and complexity.	The primary focus of DevOps is to utilize CPU-based tasks, with GPUs rarely used based on deploying ML models.	[222–225]

Table 4. *Cont.*

Aspect	LLMOps	MLOps	DevOps	References
Hardware Requirements				
TPUs (Tensor Processing Units)	TPUs can handle large parallel processing better than GPUs, making them ideal for LLMOps large models. TPUs are ideal for training LLMs at scale when speed and efficiency are crucial.	TPUs are utilized in MLOps to deploy TensorFlow-based models. They can replace GPUs in MLOps pipelines, because they are optimized for neural network matrix multiplications.	Although TPUs are not widely used in DevOps, they are still able to be utilized in high-performance circumstances that need quicker processing (deplying ML models with more computational loads).	[138,226–231]
Parallel Processing	LLMs have billions of parameters; therefore, LLMOps require considerable parallel processing (critical for large model training).	MLOps use parallel processing to speed up model training, especially in remote situations/distributed environments. Parallelism varies by model size, with many smaller ML models not requiring parallel processing.	In the context of DevOps, parallel processing is typically utilized for the purpose of managing and automating a variety of tasks, such as include testing, system monitoring, and parallel deployment pipelines.	[230,232–235]

Table 5. Comparison of LLMOps, MLOps, and DevOps.

Aspect	LLMOps	MLOps	DevOps	References
Data Management and Handling Large Datasets				
Data Volume	Handles vast datasets, frequently consisting of petabytes of textual information, to train LLMs. Highly heterogeneous, often multimodal (e.g., large-scale corpora of text, videos, etc.).	Capable of processing moderate to large datasets, adapting to the size requirements of the ML model and infrastructure. Heterogeneous data (images, text, etc.).	Manages various data volumes, focusing on operations, logs, and configuration data with a focus on cloud storage for large scale applications. (homogeneous data).	[236–238]
Data Preprocessing	Encompasses specific methodologies for processing textual data, including tokenization, contextual embeddings, and managing extensive text corpora.	Includes data cleaning, normalization, feature engineering, and transformation for diverse ML tasks (ETL tools, Pandas, Spark, etc.).	Includes system configuration and operational data preprocessing, such as log collection, normalization, and parsing for monitoring and troubleshooting.	[43,239–244]
Data Storage and Management	Due to the immense volume of text data, the utilization of modern distributed storage systems and high-throughput solutions is necessary.	Utilizes many storage technologies such as cloud storage, databases, and distributed file system to achieve scalability.	Uses cloud-based storage and databases to efficiently manage and backup operational data (e.g., AWS S3, GCS, etc.).	[145,245–249]
Data Labeling and Annotation	Usually involves the process of annotating huge amounts of text data (multimodel data annotation), typically using semi-supervised techniques or pretrained models for labeling.	Supervised learning relies on many procedures, such as manual labeling, automated annotation tools, and crowdsourcing, to ensure accurate data labeling.	Labeling operational metrics and logs can enhance analysis and monitoring, although data labeling and annotation are rare.	[250–257]
Data Versioning and Lineage	Utilizes complex versioning techniques to manage substantial amounts of training data, maintaining traceability in extensive NLP operations.	Utilized methods such as DVC (Data Version Control) to monitor modifications in datasets and guarantee reproducibility.	Uses Git for configuration management and code version control (not directly associated with data versioning).	[250,258–260]

Table 6. Comparison of LLMOps, MLOps, and DevOps.

Aspect	LLMOps	MLOps	DevOps	References
		Cloud Platforms and Services		
Primary Cloud Platforms	LLMs require specific services (training, fine tunning, etc.) and infrastructure to accommodate the extensive dataset and computational requirements. Example: AWS offers high-performance computing (HPC) services such as P3 instances and EFS (Elastic File System). Azure offers custom inferencing solutions and integration with Azure OpenAI.	Cloud providers offer a wide range of services that are well suited for various machine learning tasks, such as training models, deploying them, and monitoring their performance. Example: Amazon Web Services (AWS) provides a wide range of MLOps technologies, such as SageMaker and EFS (Elastic File System). Google Cloud Platform (GCP) encompasses the AI Platform BigQuery. Microsoft Azure offers Azure Machine Learning and Databricks.	When it comes to managing infrastructure, automation, and continuous integration and delivery pipelines, DevOps largely makes use of Amazon Web Services (EC2, S3). Google Cloud Platform (GCE, Kubernetes Engine). Microsoft Azure (App Service, Azure pipelines).	[138,261–264]
Scalability	Requires highly scalable solutions to handle the substantial computing requirements and extensive data of LLMs. Example: Highly scalable infrastructure. Distributed training refers to the process of training machine learning models over several TPUs or GPUs, such as Azure dedicated computing clusters.	Provides adaptable scaling solutions to handle diverse workloads and model specifications. Example: MLOps solutions offer adaptable scaling capabilities to accommodate a wide range of workloads. Auto-scaling services are AWS EC2 auto-scaling and Google Kubernetes Engine (GKE).	Scalable infrastructure for application deployment, automated workflows, and continuous integration and continuous delivery pipelines. It is common practice to employ auto-scaling for cloud resources Amazon Elastic Compute Cloud auto-scaling and Kubernetes.	[265–269]
Data Management and Storage	Essential for efficient LLM training, it makes use of state-of-the-art storage technologies built for high throughput and massive text datasets. Example: Particularly for LLMs, offer tailored storage solutions. Amazon Web Services: S3 for large datasets and FSX for Lustre. GCP offers BigQuery and Cloud Storage.	Data lakes, databases, and storage solutions that support a wide range of data formats. Amazon Web Services (AWS) offers three key services: S3, Redshift. Google Cloud Platform (GCP) offers several powerful data storage and processing services, including BigQuery for data analysis. Azure offers several storage options, including Data Lake Storage, Cosmos DB, and Blob Storage.	Cloud platform-based solutions such as Amazon Web Services S3 version control systems, such as Git, are essential to configuration management, since they allow for the tracking of changes. DevOps tool for data management such as DBMS, PostgreSQL, MongoDB, and MySQL.	[270–276]
Cost Management	Intense computing demands lead to increased costs; methods for cost management include using reserved instances and specialized hardware. Example: High Costs: Because LLMs use a lot of resources. Optimizing costs: Reserved Instances and Preemptible VMs. Dedicated devices like AWS Inferentia and Azure AI Accelerators are used for inference cost management.	Contains resources for monitoring and improving the efficiency of various machine learning projects budgets. Example: Tools for keeping an eye on and lowering costs are called cost-effectiveness. AWS: AWS Budgets and Cost Explorer. GCP: Tools and records for managing costs. Azure: Keeping track of costs and billing.	The management of expenses by DevOps teams is accomplished through the utilization of auto-scaling, pay-as-you-go cloud models. Tools such as Amazon Web Services, Cost Explorer, Microsoft Azure's Cost Management, and Google Clouds are frequently utilized for the purpose of monitoring and controlling expenses.	[277–280]

Table 6. *Cont.*

Aspect	LLMOps	MLOps	DevOps	References
Cloud Platforms and Services				
Service Offerings	Requires utilizing powerful computing systems for training, utilizing dedicated hardware for making predictions, and providing customized fine-tuning services. Example: High-Performance Computing (HPC) is used for training large-scale LLM models, such as Google Cloud Platform's Tensor Processing Units (TPUs), and Amazon Web Services P3 instances. Model fine-tuning can be done via APIs such as Azure OpenAI and Hugging Face on AWS.	This focuses on tools and services that facilitate the automated development, deployment, and monitoring of models. Example: AutoML Services refer to the automated process of developing and tuning models, AWS SageMaker Autopilot, and GCP AutoML. Managed Kubernetes Services, such as AWS EKS and GCP GKE, are used for deploying applications in a scalable manner. Model monitoring refers to Azure Monitor and AWS CloudWatch, to keep track of the performance and behavior of deployed models.	Continuous Integration and Continuous Delivery (CI/CD) pipelines are the focus of a wide range of services that are provided by DevOps initiatives. It is possible to use GitLab CI/CD, Jenkins, or AWS CodePipeline as references. In addition, monitoring tools such as Prometheus, Grafana, AWS CloudWatch, and Azure Monitor are essential components for monitoring the performance of the system.	[53,138,166,281–285]

6. Open Issues and Future Research Directions

This section presents a brief overview of some open issues and potential research directions:

- The integration of LLMs into Continuous Integration/Continuous Deployment (CI/CD) pipelines presents several open challenges and opportunities. Key barriers include computational costs, inaccuracies, error handling, biases, and concerns related to development, deployment, maintenance, and ethics [166]. These issues highlight the need for innovative approaches to seamlessly incorporate LLMs into CI/CD processes, ensuring they are utilized effectively and efficiently. Future research should focus on strategies to enhance the speed, reliability, and consistency of LLMs integration while mitigating associated risks and addressing the ethical implications involved.
- The evolving landscape of LLMOps presents a variety of ongoing challenges that require continued exploration. One prominent issue is the potential for LLMs to introduce inaccuracies and biases within the Continuous Integration/Continuous Deployment (CI/CD) process. This creates the need for rigorous oversight to ensure the quality and reliability of software products. Another key challenge lies in the difficulty of capturing and reproducing test scripts across diverse devices, platforms, and applications. Disparities in screen dimensions, input methods, platform functionalities, API inconsistencies, and varying application designs further complicate this issue [286]. Addressing these challenges will require innovative approaches to improve cross-platform compatibility and ensure consistent behavior of LLMs-driven systems. Future research should focus on developing strategies to mitigate biases and inaccuracies in LLMs, particularly in CI/CD workflows. Additionally, further investigation is needed into methods for standardizing test script reproduction across heterogeneous environments to enhance the scalability and reliability of LLMOps practices.
- Future research directions in LLMOps focus on advancing the development and reliability of LLMs. Key areas include integrating human feedback loops to improve model outputs, addressing biases and ethical concerns in LLMs applications, and enhancing the integration process. Additionally, there is a need to mitigate challenges

in natural language understanding and explore the potential of fine-tuning with domain-specific data to improve performance on specialized tasks. Lastly, further research is needed to formulate best practices for incorporating these models into Continuous Integration/Continuous Deployment (CI/CD) pipelines.

7. Conclusions

Generative AI, particularly LLMs, is reshaping various industries with its unparalleled capabilities in text generation and understanding. As these models continue to evolve at a rapid pace, it is crucial to comprehend the underlying architectural principles, along with the challenges and solutions required to scale these models for real-world applications. This paper highlighted the distinctions between LLMOps and MLOps, emphasizing the unique needs of LLMs in terms of deployment, monitoring, security, and scalability. By evaluating current tools, platforms, and emerging trends, we have outlined the essential infrastructure and operational strategies that practitioners must consider when managing LLMs in production environments. The rapid growth of LLMOps methodologies is key to addressing these challenges and ensuring the efficient scaling of LLMs-based applications in sectors such as healthcare, finance, and cybersecurity. Looking ahead, the evolution of LLMOps will be critical for optimizing LLMs performance and ensuring their ethical, secure, and scalable integration into diverse production environments. Thus, continued focus on these methodologies will be pivotal in advancing the future of generative AI.

Author Contributions: Conceptualization, S.P. and Z.A.; methodology, S.P.; validation, S.P.; formal analysis, S.P.; investigation, S.P.; resources, Z.A.; writing—original draft preparation, S.P.; writing—review and editing, Z.A.; visualization, S.P.; supervision, Z.A.; project administration, Z.A.; funding acquisition, Z.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data sharing is not applicable.

Conflicts of Interest: Saurabh Pahune was employed by the company Cardinal Health, Dublin, OH 43017, USA.

Abbreviations

The following abbreviations are used in this manuscript:

LLMs	Large Language Models
LLMOps	Large Language Model Operations
MLOps	Machine Learning Operations
BERT	Bidirectional Encoder Representations from Transformers
GPT	Generative Pretrained Transformer
LLaMA	Large Language Model Meta AI
GenAI	Generative Artificial Intelligence
AIOps	Artificial Intelligence for IT Operations
RAGOps	Retrieval-Augmented Generation Operations
GenAIOps	Generative AI Operations
T5	Text-to-Text Transfer Transformer

AutoML	Automated Machine Learning
YOLO	You Only Look Once
RAG	Retrieval-Augmented Generation
LSTM	Long Short-Term Memory
DevOps	Development and Operations
CI/CD	Continuous Integration and Continuous Delivery
DataOps	Data Operations
SDLC	Software Development Life Cycle
CSFs	Critical Success Factors
MLEs	Machine Learning Engineers
FMOps	Foundation Model Operations
HITL	Human-in-the-Loop
LLMCI	LLM–Computer Interaction
GCP	Google Cloud Platform
AWS	Amazon Web Services
EC2	Elastic Compute Cloud
S3	Simple Storage Service
NLP	Natural Language Processing
ETL	Extract, Transform, and Load
ZeRO	Zero Redundancy Optimizer
DevSecOps	Development, Security, and Operations
GPU	Graphics Processing Unit
TPU	Tensor Processing Unit
GCS	Google Cloud Storage
DVC	Data Version Control
EFS	Elastic File System
HPC	High-Performance Computing
GKE	Google Kubernetes Engine
UI	User Interfaces
AI	Artificial Intelligence

References

1. Eboigbe, E.O.; Farayola, O.A.; Olatoye, F.O.; Nnabugwu, O.C.; Daraojimba, C. Business intelligence transformation through AI and Data Analytics. *Eng. Sci. Technol. J.* **2023**, *4*, 285–307. [CrossRef]
2. Ghobakhloo, M.; Fathi, M.; Iranmanesh, M.; Vilkas, M.; Grybauskas, A.; Amran, A. Generative artificial intelligence in manufacturing: Opportunities for actualizing Industry 5.0 sustainability goals. *J. Manuf. Technol. Manag.* **2024**, *35*, 94–121. [CrossRef]
3. Szmurlo, H.; Akhtar, Z. Digital Sentinels and Antagonists: The Dual Nature of Chatbots in Cybersecurity. *Information* **2024**, *15*, 443. [CrossRef]
4. Zhang, P.; Kamel Boulos, M.N. Generative AI in medicine and healthcare: Promises, opportunities and challenges. *Future Internet* **2023**, *15*, 286. [CrossRef]
5. Pahune, S. Large Language Models and Generative AI's Expanding Role in Healthcare. Available online: https://www.researchgate.net/publication/377217911_Large_Language_Models_and_Generative_AI's_Expanding_Role_in_Healthcare (accessed on 22 September 2024).
6. EY Insights. How Generative AI in Supply Chain Can Drive Value. 2023. Available online: https://www.ey.com/en_us/insights/supply-chain/how-generative-ai-in-supply-chain-can-drive-value (accessed on 22 September 2024).
7. Jackson, I.; Ivanov, D.; Dolgui, A.; Namdar, J. Generative artificial intelligence in supply chain and operations management: A capability-based framework for analysis and implementation. *Int. J. Prod. Res.* **2024**, *62*, 6120–6145. [CrossRef]
8. Ebert, C.; Louridas, P. Generative AI for software practitioners. *IEEE Softw.* **2023**, *40*, 30–38. [CrossRef]
9. Akpınar, M.T. Generative Artificial Intelligence Applications Specific to the Air Transport Industry. In *Interdisciplinary Studies on Contemporary Research Practices in Engineering in the 21st Century II*; Kaygusuz, K., Ed.; Özgür Publications: İstanbul, Turkey, 2023. [CrossRef]
10. InData Labs. AI Latest Developments. 2023. Available online: https://indatalabs.com/blog/ai-latest-developments (accessed on 17 August 2024).

11. Ajiga, D.; Okeleke, P.A.; Folorunsho, S.O.; Ezeigweneme, C. The role of software automation in improving industrial operations and efficiency. *Int. J. Eng. Res. Update* **2024**, *7*, 22–35. [CrossRef]
12. Schwartz, R.; Schwartz, R.; Vassilev, A.; Greene, K.; Perine, L.; Burt, A.; Hall, P. *Towards a Standard for Identifying and Managing Bias in Artificial Intelligence*; US Department of Commerce, National Institute of Standards and Technology: Gaithersburg, MD, USA, 2022; Volume 3.
13. Song, C.; Raghunathan, A. Information leakage in embedding models. In Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security, Virtual, 9–13 November 2020; pp. 377–390.
14. Hitaj, B.; Ateniese, G.; Perez-Cruz, F. Deep models under the GAN: Information leakage from collaborative deep learning. In Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security, Dallas, TX, USA, 30 October–3 November 2017; pp. 603–618.
15. Rigaki, M.; Garcia, S. A survey of privacy attacks in machine learning. *ACM Comput. Surv.* **2023**, *56*, 1–34. [CrossRef]
16. Li, P.; Wang, X.; Huang, K.; Huang, Y.; Li, S.; Iqbal, M. Multi-model running latency optimization in an edge computing paradigm. *Sensors* **2022**, *22*, 6097. [CrossRef]
17. Greco, S.; Vacchetti, B.; Apiletti, D.; Cerquitelli, T. Unsupervised Concept Drift Detection from Deep Learning Representations in Real-time. *arXiv* **2024**, arXiv:2406.17813.
18. Kreuzberger, D.; Kühl, N.; Hirschl, S. Machine learning operations (mlops): Overview, definition, and architecture. *IEEE Access* **2023**, *11*, 31866–31879. [CrossRef]
19. Symeonidis, G.; Nerantzis, E.; Kazakis, A.; Papakostas, G.A. Mlops-definitions, tools and challenges. In Proceedings of the 2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC), Virtual, 26–29 January 2022; pp. 453–460.
20. John Snow Labs. Introduction to Large Language Models (LLMs): An Overview of BERT, GPT, and Other Popular Models. 2024. Available online: https://www.johnsnowlabs.com/introduction-to-large-language-models-llms-an-overview-of-bert-gpt-and-other-popular-models/ (accessed on 14 September 2024).
21. Gao, Y.; Baptista-Hon, D.T.; Zhang, K. The inevitable transformation of medicine and research by large language models: The possibilities and pitfalls. *MEDCOMM-Future Med.* **2023**, *2*, 1–2. [CrossRef]
22. Minaee, S.; Mikolov, T.; Nikzad, N.; Chenaghlu, M.; Socher, R.; Amatriain, X.; Gao, J. Large language models: A survey. *arXiv* **2024**, arXiv:2402.06196.
23. Zhu, Y.; Yuan, H.; Wang, S.; Liu, J.; Liu, W.; Deng, C.; Chen, H.; Dou, Z.; Wen, J.R. Large language models for information retrieval: A survey. *arXiv* **2023**, arXiv:2308.07107.
24. Lee, J.; Stevens, N.; Han, S.C.; Song, M. A survey of large language models in finance (finllms). *arXiv* **2024**, arXiv:2402.02315.
25. Yuan, F.; Yuan, S.; Wu, Z.; Li, L. How Multilingual is Multilingual LLM? *arXiv* **2023**, arXiv:2311.09071.
26. Dada, A.; Bauer, M.; Contreras, A.B.; Koraş, O.A.; Seibold, C.M.; Smith, K.E.; Kleesiek, J. CLUE: A Clinical Language Understanding Evaluation for LLMs. *arXiv* **2024**, arXiv:2404.04067.
27. Wang, W.; Chen, Z.; Chen, X.; Wu, J.; Zhu, X.; Zeng, G.; Luo, P.; Lu, T.; Zhou, J.; Qiao, Y.; et al. Visionllm: Large language model is also an open-ended decoder for vision-centric tasks. In Proceedings of the 37th Conference on Neural Information Processing Systems, New Orleans, LA, USA, 6–10 December 2024.
28. Naveed, H.; Khan, A.U.; Qiu, S.; Saqib, M.; Anwar, S.; Usman, M.; Akhtar, N.; Barnes, N.; Mian, A. A comprehensive overview of large language models. *arXiv* **2023**, arXiv:2307.06435.
29. Pahune, S.; Chandrasekharan, M. Several categories of large language models (llms): A short survey. *arXiv* **2023**, arXiv:2307.10188. [CrossRef]
30. Hadi, M.U.; Qureshi, R.; Shah, A.; Irfan, M.; Zafar, A.; Shaikh, M.B.; Akhtar, N.; Wu, J.; Mirjalili, S.; Shah, M.; et al. Large language models: A comprehensive survey of its applications, challenges, limitations, and future prospects. *Authorea Prepr.* **2023**. [CrossRef]
31. Zheng, J.; Qiu, S.; Shi, C.; Ma, Q. Towards Lifelong Learning of Large Language Models: A Survey. *arXiv* **2024**, arXiv:2406.06391.
32. Huang, K.; Wang, Y.; Goertzel, B.; Li, Y.; Wright, S.; Ponnapalli, J. *Generative AI Security*; Springer: Berlin/Heidelberg, Germany, 2024. [CrossRef]
33. Yao, Y.; Duan, J.; Xu, K.; Cai, Y.; Sun, Z.; Zhang, Y. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confid. Comput.* **2024**, *4*, 100211. [CrossRef]
34. Zhao, S.; Tuan, L.A.; Fu, J.; Wen, J.; Luo, W. Exploring Clean Label Backdoor Attacks and Defense in Language Models. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2024**, *32*, 3014–3024. [CrossRef]
35. Yao, J.; Luo, H.; Zhang, X.L. Interpretable spectrum transformation attacks to speaker recognition. *arXiv* **2023**, arXiv:2302.10686. [CrossRef]
36. Tong, C.; Zheng, X.; Li, J.; Ma, X.; Gao, L.; Xiang, Y. Query-Efficient Black-Box Adversarial Attacks on Automatic Speech Recognition. *IEEE/ACM Trans. Audio Speech Lang. Process.* **2023**, *31*, 3981–3992. [CrossRef]
37. Alber, D.A.; Yang, Z.; Alyakin, A.; Yang, E.; Rai, S.; Valliani, A.A.; Zhang, J.; Rosenbaum, G.R.; Amend-Thomas, A.K.; Kurland, D.B.; et al. Medical large language models are vulnerable to data-poisoning attacks. *Nat. Med.* **2025**, 1–9. [CrossRef]

38. Chu, J.; Liu, Y.; Yang, Z.; Shen, X.; Backes, M.; Zhang, Y. Comprehensive assessment of jailbreak attacks against llms. *arXiv* **2024**, arXiv:2402.05668.
39. Abdali, S.; He, J.; Barberan, C.; Anarfi, R. Can LLMs be Fooled? Investigating Vulnerabilities in LLMs. *arXiv* **2024**, arXiv:2407.20529.
40. Signoz. LLMOps: What Is LLMOps and How Is It Different from MLOps? 2023. Available online: https://signoz.io/guides/llmops/ (accessed on 18 August 2024).
41. Shan, R.; Shan, T. Enterprise LLMOps: Advancing Large Language Models Operations Practice. In Proceedings of the 2024 IEEE Cloud Summit, Washington, DC, USA, 27–28 June 2024; pp. 143–148.
42. Kamath, U.; Keenan, K.; Somers, G.; Sorenson, S. LLMs in Production. In *Large Language Models: A Deep Dive: Bridging Theory and Practice*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 315–373.
43. Chenxi, S.; Liang, P.; Wu, Y.; Zhan, T.; Jin, Z. Maximizing user experience with LLMOps-driven personalized recommendation systems. *Appl. Comput. Eng.* **2024**, *64*, 100–106. [CrossRef]
44. AIMultiple Research. LLM Security: Ensuring Safe and Secure Use of Large Language Models. 2023. Available online: https://research.aimultiple.com/llm-security/ (accessed on 18 August 2024).
45. Najafabadi, F.A.; Bogner, J.; Gerostathopoulos, I.; Lago, P. An Analysis of MLOps Architectures: A Systematic Mapping Study. *arXiv* **2024**, arXiv:2406.19847.
46. Faubel, L.; Schmid, K. MLOps: A Multiple Case Study in Industry 4.0. *arXiv* **2024**, arXiv:2407.09107.
47. di Laurea, I.S. Mlops-Standardizing the Machine Learning Workflow. Ph.D. Thesis, University of Bologna, Bologna, Italy, 2021.
48. Xu, R. A Design Pattern for Deploying Machine Learning Models to Production. 2020. Available online: https://scholarworks.calstate.edu/downloads/1v53k296v (accessed on 29 October 2024).
49. Eken, B.; Pallewatta, S.; Tran, N.K.; Tosun, A.; Babar, M.A. A Multivocal Review of MLOps Practices, Challenges and Open Issues. *arXiv* **2024**, arXiv:2406.09737.
50. Testi, M.; Ballabio, M.; Frontoni, E.; Iannello, G.; Moccia, S.; Soda, P.; Vessio, G. MLOps: A taxonomy and a methodology. *IEEE Access* **2022**, *10*, 63606–63618. [CrossRef]
51. Jana, A.D. The MLOps Approach to Model Deployment: A Road Map to Seamless Scalability. *J. Artif. Intell. Cloud Comput.* **2022**, *1*, 1–4. [CrossRef]
52. Matsui, B.M.A.; Goya, D.H. MLOps: Five Steps to Guide its Effective Implementation. In Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI, Pittsburgh, PA, USA, 16–17 May 2022. [CrossRef]
53. Tabassam, A. MLOps: A Step Forward to Enterprise Machine Learning. *arXiv* **2023**. [CrossRef]
54. Wazir, S.; Kashyap, G.S.; Saxena, P. Mlops: A review. *arXiv* **2023**, arXiv:2308.10908.
55. Barring, N. AIOps vs MLOps vs LLMOps: Choosing the Right AI Operations Strategy. 2024. Available online: https://www.nscale.com/blog/aiops-vs-mlops-vs-llmops-choosing-the-right-ai-operations-strategy (accessed on 28 December 2024).
56. Prabhune, S.; Berndt, D.J. Deploying Large Language Models with Retrieval Augmented Generation. *arXiv* **2024**. [CrossRef]
57. NVIDIA. Mastering LLM Techniques with LLMOps. 2023. Available online: https://developer.nvidia.com/blog/mastering-llm-techniques-llmops/ (accessed on 26 December 2024).
58. Gupta, P.; Bagchi, A. MLOps: Machine Learning Operations. In *Essentials of Python for Artificial Intelligence and Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 489–518.
59. Databricks. MLOps Glossary. 2024. Available online: https://www.databricks.com/solutions/machine-learning (accessed on 22 September 2024).
60. Nogare, D.; Silveira, I.F. Experimentation, deployment and monitoring Machine Learning models: Approaches for applying MLOps. *arXiv* **2024**, arXiv:2408.11112.
61. Databricks. MLOps. 2023. Available online: https://www.databricks.com/glossary/mlops (accessed on 18 August 2024).
62. Ahmad, T.; Adnan, M.; Rafi, S.; Akbar, M.A.; Anwar, A. MLOps-Enabled Security Strategies for Next-Generation Operational Technologies. In Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering, Salerno, Italy, 18–21 June 2024; pp. 662–667.
63. Liang, P.; Song, B.; Zhan, X.; Chen, Z.; Yuan, J. Automating the Training and Deployment of Models in MLOps by Integrating Systems with Machine Learning. *arXiv* **2024**, arXiv:2405.09819.
64. Subramanya, R.; Sierla, S.; Vyatkin, V. From DevOps to MLOps: Overview and application to electricity market forecasting. *Appl. Sci.* **2022**, *12*, 9851. [CrossRef]
65. Banerjee, A.; Chen, C.C.; Hung, C.C.; Huang, X.; Wang, Y.; Chevesaran, R. Challenges and Experiences with {MLOps} for Performance Diagnostics in {Hybrid-Cloud} Enterprise Software Deployments. In Proceedings of the 2020 USENIX Conference on Operational Machine Learning (OpML 20), Santa Clara, CA, USA, 28 July–7 August 2020.
66. Neptune.ai. MLOps: A Comprehensive Guide to Machine Learning Operations. 2023. Available online: https://neptune.ai/blog/mlops (accessed on 18 August 2024).

67. Mehmood, Y.; Sabahat, N.; Ijaz, M.A. MLOps critical success factors—A systematic literature review. *VFAST Trans. Softw. Eng.* **2024**, *12*, 183–209. [CrossRef]
68. Joshi, A. MLOps Mastery: Streamlining Machine Learning Lifecycle Management. *Int. J. Sci. Res.* **2024**. [CrossRef]
69. Kabbay, H.S. Streamlining AI Application: MLOps Best Practices and Platform Automation Illustrated through an Advanced RAG based Chatbot. In Proceedings of the 2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS), Coimbatore, India, 10–12 July 2024. [CrossRef]
70. Bodor, A.; Hnida, M.; Daoudi, N. From Development to Deployment: An Approach to MLOps Monitoring for Machine Learning Model Operationalization. In Proceedings of the 2023 14th International Conference on Intelligent Systems: Theories and Applications (SITA), Casablanca, Morocco, 22–23 November 2023. [CrossRef]
71. Godwin, R.C.; Melvin, R.L. Toward efficient data science: A comprehensive MLOps template for collaborative code development and automation. *SoftwareX* **2024**, *26*, 101723. [CrossRef]
72. Tatineni, S.; Boppana, V.R. AI-Powered DevOps and MLOps Frameworks: Enhancing Collaboration, Automation, and Scalability in Machine Learning Pipelines. *J. Artif. Intell. Res. Appl.* **2021**, *1*, 58–88.
73. Sothilingam, R.; Pant, V.; Eric, S. Using i* to Analyze Collaboration Challenges in MLOps Project Teams. In Proceedings of the iStar, Hyderabad, India, 17 October 2022; pp. 1–6.
74. Torres, A.P.G.; Sawhney, N. Role of Regulatory Sandboxes and MLOps for AI-Enabled Public Sector Services. *Rev. Socionetwork Strateg.* **2023**, *17*, 297–318. [CrossRef]
75. Pulicharla, M.R. Data Versioning and Its Impact on Machine Learning Models. *J. Sci. Technol.* **2024**, *5*, 22–37. [CrossRef]
76. Makinen, S.; Skogstrom, H.; Laaksonen, E.; Mikkonen, T. Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help? In Proceedings of the 2021 IEEE/ACM 1st Workshop on AI Engineering—Software Engineering for AI (WAIN), Madrid, Spain, 30–31 May 2021. [CrossRef]
77. Narayanappa, A.K.; Amrit, C. An Analysis of the Barriers Preventing the Implementation of MLOps. In Proceedings of the IFIP Advances in Information and Communication Technology, Denton, TX, USA, 2–3 November 2023. [CrossRef]
78. Prasanna, G. Optimizing the Future: Unveiling the Significance of MLOps in Streamlining the Machine Learning Lifecycle. *Int. J. Sci. Res. Eng. Technol.* **2024**, *4*, 5–8. [CrossRef]
79. Gallinucci, E. MLOps—Standardizing the Machine Learning Workflow. 2023. Available online: https://amslaurea.unibo.it/id/eprint/23645/1/tesi_enrico_salvucci.pdf (accessed on 22 December 2024).
80. Varga, P.; Kővári, Á.; Herkules, M.; Hegedűs, C. MLOps in CPS–a use-case for image recognition in changing industrial settings. In Proceedings of the NOMS 2024—2024 IEEE Network Operations and Management Symposium, Seoul, Republic of Korea, 6–10 May 2024; pp. 1–4.
81. Araujo, G.; Kalinowski, M.; Endler, M.; Calefato, F. Professional Insights into Benefits and Limitations of Implementing MLOps Principles. *arXiv* **2024**, arXiv:2403.13115.
82. Tembhekar, P.; Malaiyappan, J.N.A.; Shanmugam, L. Cross-Domain Applications of MLOps: From Healthcare to Finance. *J. Knowl. Learn. Sci. Technol.* **2023**, *2*, 581–598. [CrossRef]
83. Moskalenko, V.; Kharchenko, V. Resilience-aware MLOps for AI-based medical diagnostic system. *Front. Public Health* **2024**, *12*, 1342937. [CrossRef] [PubMed]
84. Sitcheu, A.Y.; Friederich, N.; Baeuerle, S.; Neumann, O.; Reischl, M.; Mikut, R. MLOps for Scarce Image Data: A Use Case in Microscopic Image Analysis. In Proceedings of the Proceedings-33. Workshop Computational Intelligence, Berlin, Germany, 23–24 November 2023; KIT Scientific Publishing: Karlsruhe, Germany, 2023; Volume 23, p. 169.
85. Reddy, M.; Dattaprakash, B.; Kammath, S.; Kn, S.; Manokaran, S.; Be, R. Application of MLOps in Prediction of Lifestyle Diseases. *ECS Trans.* **2022**, *107*, 1191. [CrossRef]
86. Vartak, M. From ml models to intelligent applications: The rise of mlops. *Proc. Vldb Endow.* **2021**, *14*, 3419. [CrossRef]
87. Posoldova, A. Machine learning pipelines: From research to production. *IEEE Potentials* **2020**, *39*, 38–42. [CrossRef]
88. Silva, L.C.; Zagatti, F.R.; Sette, B.S.; dos Santos Silva, L.N.; Lucrédio, D.; Silva, D.F.; de Medeiros Caseli, H. Benchmarking machine learning solutions in production. In Proceedings of the 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), Miami, FL, USA, 14–17 December 2020; pp. 626–633.
89. Karlaš, B.; Interlandi, M.; Renggli, C.; Wu, W.; Zhang, C.; Mukunthu Iyappan Babu, D.; Edwards, J.; Lauren, C.; Xu, A.; Weimer, M. Building continuous integration services for machine learning. In Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, Virtual Event CA USA, 6–10 July 2020; pp. 2407–2415.
90. Nigenda, D.; Karnin, Z.; Zafar, M.B.; Ramesha, R.; Tan, A.; Donini, M.; Kenthapadi, K. Amazon sagemaker model monitor: A system for real-time insights into deployed machine learning models. In Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Washington, DC, USA, 14–18 August 2022; pp. 3671–3681.
91. Karakus, C.; Huilgol, R.; Wu, F.; Subramanian, A.; Daniel, C.; Cavdar, D.; Xu, T.; Chen, H.; Rahnama, A.; Quintela, L. Amazon sagemaker model parallelism: A general and flexible framework for large model training. *arXiv* **2021**, arXiv:2111.05972.

92. Das, P.; Ivkin, N.; Bansal, T.; Rouesnel, L.; Gautier, P.; Karnin, Z.; Dirac, L.; Ramakrishnan, L.; Perunicic, A.; Shcherbatyi, I.; et al. Amazon SageMaker Autopilot: A white box AutoML solution at scale. In Proceedings of the Fourth International Workshop on Data Management for End-to-End Machine Learning, Portland, OR, USA, 14 June 2020; pp. 1–7.
93. Choi, W.; Choi, T.; Heo, S. A Comparative Study of Automated Machine Learning Platforms for Exercise Anthropometry-Based Typology Analysis: Performance Evaluation of AWS SageMaker, GCP VertexAI, and MS Azure. *Bioengineering* **2023**, *10*, 891. [CrossRef] [PubMed]
94. Bagai, R. Comparative Analysis of AWS Model Deployment Services. *arXiv* **2024**, arXiv:2405.08175.
95. Pala, S.K. Databricks Analytics: Empowering Data Processing, Machine Learning and Real-Time Analytics. *Mach. Learn.* **2021**, *10*, 76–82.
96. L'Esteve, R. Databricks. In *The Azure Data Lakehouse Toolkit*; Apress: Berkeley, CA, USA, 2022. [CrossRef]
97. Althati, C.; Tomar, M.; Malaiyappan, J.N.A. Scalable Machine Learning Solutions for Heterogeneous Data in Distributed Data Platform. *J. Artif. Intell. Gen. Sci.* **2024**, *4*, 299–309. [CrossRef]
98. Ruan, W.; Chen, Y.; Forouraghi, B. On Development of Data Science and Machine Learning Applications in Databricks. In Proceedings of the World Congress on Services, San Diego, CA, USA, 25–30 June 2019; Springer: Berlin/Heidelberg, Germany, 2019; pp. 78–91.
99. Zaharia, M. Lessons from large-scale software as a service at databricks. In Proceedings of the ACM Symposium on Cloud Computing, Santa Cruz, CA, USA, 20–23 November 2019; p. 101.
100. El Moutaouakal, W.; Baïna, K. Comparative Experimentation of MLOps Power on Microsoft Azure, Amazon Web Services, and Google Cloud Platform. In Proceedings of the 2023 IEEE 6th International Conference on Cloud Computing and Artificial Intelligence: Technologies and Applications (CloudTech), Marrakesh, Morocco, 21–23 November 2023; pp. 1–8.
101. De Rosa, P.; Bromberg, Y.D.; Felber, P.; Mvondo, D.; Schiavoni, V. On the Cost of Model-Serving Frameworks: An Experimental Evaluation. In Proceedings of the 2024 IEEE International Conference on Cloud Engineering (IC2E), Paphos, Cyprus, 24–27 September 2024; pp. 221–232.
102. PyTorch Team. PyTorch. Available online: https://pytorch.org/ (accessed on 6 October 2024).
103. Hao, Y.; Zhao, X.; Bao, B.; Berard, D.; Constable, W.; Aziz, A.; Liu, X. Torchbench: Benchmarking pytorch with high api surface coverage. *arXiv* **2023**, arXiv:2304.14226.
104. Mishra, P. Distributed PyTorch Modelling, Model Optimization, and Deployment. In *PyTorch Recipes: A Problem-Solution Approach to Build, Train and Deploy Neural Network Models*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 187–212.
105. Jain, S.; Kumar, P. Cost Effective Generic Machine Learning Operation: A Case Study. In Proceedings of the 2023 International Conference on Data Science and Network Security (ICDSNS), Tiptur, India, 28–29 July 2023; pp. 1–6.
106. Documentation, M. MLflow Tutorials and Examples. Available online: https://mlflow.org/docs/latest/tutorials-and-examples/index.html (accessed on 6 October 2024).
107. Hsu, C.C.; Chen, P.H.; Wu, I.Z. End-to-End Automation of ML Model Lifecycle Management using Machine Learning Operations Platforms. In Proceedings of the 2024 International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan), Kaohsiung, Taiwan, 16–18 July 2024; pp. 209–210.
108. Vijayan, N.E. Building Scalable MLOps: Optimizing Machine Learning Deployment and Operations. *Indian Sci. J. Res. Eng. Manag.* **2024**, *8*, 1–9. [CrossRef]
109. Heydari, M.; Rezvani, Z. Challenges and Experiences of Iranian Developers with MLOps at Enterprise. In Proceedings of the 2023 7th Iranian Conference on Advances in Enterprise Architecture (ICAEA), Tehran, Iran, 15–16 November 2023. [CrossRef]
110. Zhou, Y.; Yu, Y.; Ding, B. Towards mlops: A case study of ml pipeline platform. In Proceedings of the 2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE), Beijing, China, 22–25 October 2020; pp. 494–500.
111. Yuan, D.Y.; Wildish, T. Bioinformatics application with kubeflow for batch processing in clouds. In Proceedings of the International Conference on High Performance Computing, Pune, India, 16–19 December 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 355–367.
112. Tagliabue, J.; Bowne-Anderson, H.; Tuulos, V.; Goyal, S.; Cledat, R.; Berg, D. Reasonable scale machine learning with open-source metaflow. *arXiv* **2023**, arXiv:2303.11761.
113. Documentation, M. What Is Metaflow? Available online: https://docs.metaflow.org/introduction/what-is-metaflow (accessed on 6 October 2024).
114. InfoQ. Netflix Introduces Metaflow: Simplifying Human-Centric AI Development. 2024. Available online: https://www.infoq.com/news/2024/03/netflix-metaflow/#:~:text=By%20creating%20various%20integrations%20for,without%20incurring%20unsustainable%20operational%20overhead (accessed on 6 October 2024).
115. Gliozzo, A.; Biran, O.; Patwardhan, S.; McKeown, K. Semantic technologies in IBM Watson. In Proceedings of the Fourth Workshop on Teaching NLP and CL, Sofia, Bulgaria, 9 August 2013; pp. 85–92.

116. Packowski, S.; Lakhana, A. Using IBM watson cloud services to build natural language processing solutions to leverage chat tools. In Proceedings of the 27th Annual International Conference on Computer Science and Software Engineering, Markham, ON, Canada, 6–8 November 2017; pp. 211–218.
117. Cloudera. Cloudera Delivers Open Standards-Based MLOps, Empowering Enterprises to Industrialize AI. 2020. Available online: https://www.cloudera.com/about/news-and-blogs/press-releases/2020-05-06-cloudera-delivers-open-standards-based-mlops-empowering-enterprises-to-industrialize-ai.html (accessed on 6 October 2024).
118. Valova, I. Research and Analysis of the Execution of Different Types of SQL Queries with Impala in Cloudera in Education. In Proceedings of the 2024 XXXIII International Scientific Conference Electronics (ET), Sozopol, Bulgaria, 17–19 September 2024; pp. 1–5.
119. Martins, R.R. Automation of Machine Learning Pipelines for Anomaly Detection Challenges. Ph.D. Thesis, Universidade do Minho, Braga, Portugal, 2023.
120. Contino. Apache Airflow: The Hands-On Guide. Available online: https://www.contino.io/insights/apache-airflow (accessed on 6 October 2024).
121. Yasmin, J.; Wang, J.A.; Tian, Y.; Adams, B. An empirical study of developers' challenges in implementing Workflows as Code: A case study on Apache Airflow. *J. Syst. Softw.* **2025**, *219*, 112248. [CrossRef]
122. Tian, L.; Sedona, R.; Mozaffari, A.; Kreshpa, E.; Paris, C.; Riedel, M.; Schultz, M.G.; Cavallaro, G. End-to-End Process Orchestration of Earth Observation Data Workflows with Apache Airflow on High Performance Computing. In Proceedings of the IGARSS 2023—2023 IEEE International Geoscience and Remote Sensing Symposium, Pasadena, CA, USA, 16–21 July 2023; pp. 711–714.
123. Abdelkader, H.; Abdelrazek, M.; Schneider, J.G.; Rani, P.; Vasa, R. Robustness Attributes to Safeguard Machine Learning Models in Production. In Proceedings of the 2023 IEEE Engineering Informatics, Melbourne, Australia, 22–23 November 2023; pp. 1–9.
124. Amershi, S.; Begel, A.; Bird, C.; DeLine, R.; Gall, H.; Kamar, E.; Nagappan, N.; Nushi, B.; Zimmermann, T. Software engineering for machine learning: A case study. In Proceedings of the 2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP), Montreal, QC, Canada, 27 May 2019; pp. 291–300.
125. Polyzotis, N.; Roy, S.; Whang, S.E.; Zinkevich, M. Data management challenges in production machine learning. In Proceedings of the 2017 ACM International Conference on Management of Data, Chicago, IL, USA, 14–19 May 2017; pp. 1723–1726.
126. Haertel, C.; Staegemann, D.; Daase, C.; Pohl, M.; Nahhas, A.; Turowski, K. MLOps in Data Science Projects: A Review. In Proceedings of the 2023 IEEE International Conference on Big Data (BigData), Sorrento, Italy, 15–18 December 2023; pp. 2396–2404.
127. de Arcaya, J.D.; de Armentia, J.L.; Miñón, R.; Ojanguren, I.L.; Torre-Bastida, A.I. Large Language Model Operations (LLMOps): Definition, Challenges, and Lifecycle Management. In Proceedings of the 2024 9th International Conference on Smart and Sustainable Technologies (SpliTech), Bol and Split, Croatia, 25–28 June 2024. [CrossRef]
128. Weights & Biases. Understanding LLMOps: Large Language Model Operations. Available online: https://wandb.ai/site/articles/understanding-llmops-large-language-model-operations/ (accessed on 7 September 2024).
129. Sinha, M.; Menon, S.; Sagar, R. LLMOps: Definitions, Framework and Best Practices. In Proceedings of the 2024 International Conference on Electrical, Computer and Energy Technologies (ICECET), Sydney, Australia, 25–27 July 2024; pp. 1–6. [CrossRef]
130. Huang, K.; Manral, V.; Wang, W. *From LLMOps to DevSecOps for GenAI*; Springer: Berlin/Heidelberg, Germany, 2023. [CrossRef]
131. Databricks. The Big Book of MLOps—2nd Edition. 2023. Available online: https://www.databricks.com/sites/default/files/2024-06/2023-10-EB-Big-Book-of-MLOps-2nd-Edition.pdf (accessed on 5 October 2024).
132. Kolesnikov, D.; Kozlova, A.; Alexandrov, A.; Kalmykov, N.; Treshkov, P.; LeBaron, T.W.; Medvedev, O. Applying ChatGPT to writing scientific articles on the use of telemedicine: Opportunities and limitations. *Artif. Intell. Health* **2024**, *1*, 53–63. [CrossRef]
133. Korada, L. GitHub Copilot: The Disrupting AI Companion Transforming the Developer Role and Application Lifecycle Management. *J. Artif. Intell. Cloud Comput.* **2024**, *3*, 1–4. [CrossRef]
134. Wermelinger, M. *Using GitHub Copilot to Solve Simple Programming Problems*; Association for Computing Machinery: New York, NY, USA, 2023. [CrossRef]
135. Osawa, K. Integrating Automated Written Corrective Feedback into E-Portfolios for second language Writing: Notion and Notion AI. *RELC J.* **2023**, *55*. [CrossRef]
136. Jasper AI. Jasper AI. 2024. Available online: https://www.jasper.ai/ (accessed on 28 September 2024).
137. Chang, T.A.; Bergen, B.K. Language model behavior: A comprehensive survey. *Comput. Linguist.* **2024**, *50*, 293–350. [CrossRef]
138. Duan, J.; Zhang, S.; Wang, Z.; Jiang, L.; Qu, W.; Hu, Q.; Wang, G.; Weng, Q.; Yan, H.; Zhang, X.; et al. Efficient Training of Large Language Models on Distributed Infrastructures: A Survey. *arXiv* **2024**. [CrossRef]
139. Lee, Y.S.; Lee, J.K. A Study on Technological Perception Analysis of LLMs through Big Data Analysis of News Articles. *J. Korea Multimed. Soc.* **2024**, *27*, 287–298. [CrossRef]
140. Mailach, A.; Simon, S.; Dorn, J.; Siegmund, N. Practitioners' Discussions on Building LLM-based Applications for Production. *arXiv* **2024**, arXiv:2411.08574.
141. Park, C.; Jiang, J.; Wang, S.; Paul, S.; Tang, J. LlamaDuo: LLMOps Pipeline for Seamless Migration from Service LLMs to Small-Scale Local LLMs. *arXiv* **2024**. [CrossRef]

142. Abdelnabi, S.; Fay, A.; Cherubin, G.; Salem, A.; Fritz, M.; Paverd, A. Are you still on track!? Catching LLM Task Drift with Activations. *arXiv* **2024**. [CrossRef]
143. Echterhoff, J.; Faghri, F.; Vemulapalli, R.; Hu, T.Y.; Li, C.; Tuzel, O.; Pouransari, H. MUSCLE: A Model Update Strategy for Compatible LLM Evolution. *arXiv* **2024**. [CrossRef]
144. Bhan, L. Mastering LLMOps: Best Practices for Managing and Deploying Large Language Models. 2023. Available online: https://lekha-bhan88.medium.com/mastering-llmops-best-practices-for-managing-and-deploying-large-language-models-c8ca0da648d9 (accessed on 26 December 2024).
145. Fernandez, R.C.; Elmore, A.J.; Franklin, M.J.; Krishnan, S.; Tan, C. How large language models will disrupt data management. *Proc. VLDB Endow.* **2023**, *16*, 3302–3309. [CrossRef]
146. Snell, C.; Lee, J.; Xu, K.; Kumar, A. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv* **2024**, arXiv:2408.03314.
147. Leiker, D.; Finnigan, S.; Gyllen, A.R.; Cukurova, M. Prototyping the use of Large Language Models (LLMs) for adult learning content creation at scale. *arXiv* **2023**, arXiv:2306.01815.
148. Hassani, S. Enhancing Legal Compliance and Regulation Analysis with Large Language Models. *arXiv* **2024**, arXiv:2404.17522.
149. Soni, V. Large language models for enhancing customer lifecycle management. *J. Empir. Soc. Sci. Stud.* **2023**, *7*, 67–89.
150. Preuss, N.; Alshehri, A.S.; You, F. Large Language Models for Life Cycle Assessments: Opportunities, Challenges, and Risks. *J. Clean. Prod.* **2024**, *466*, 142824. [CrossRef]
151. Yosifova, V. Application of Open- source Large Language Model (LLM) for Simulation of a Vulnerable IoT System and Cybersecurity Best Practices Assistance. *Preprints* **2024**. [CrossRef]
152. Xu, J.; Stokes, J.W.; McDonald, G.; Bai, X.; Marshall, D.; Wang, S.; Swaminathan, A.; Li, Z. AutoAttacker: A Large Language Model Guided System to Implement Automatic Cyber-attacks. *arXiv* **2024**. [CrossRef]
153. Vinayak, E.S.; Anbuthiruvarangan, M.K.; Chakradhar, K.; P, A. Enhancing Cybersecurity Through AI-Driven Threat Detection: A Transfer Learning Approach. *Int. J. Multidiscip. Res.* **2024**, 6. [CrossRef]
154. Hassanin, M.; Keshk, M.; Salim, S.; Alsubaie, M.; Sharma, D. PLLM-CS: Pre-trained Large Language Model (LLM) for Cyber Threat Detection in Satellite Networks. *arXiv* **2024**. [CrossRef]
155. Boi, B.; Esposito, C.; Lee, S. Smart Contract Vulnerability Detection: The Role of Large Language Model (LLM). *ACM SIGAPP Appl. Comput. Rev.* **2024**, *24*, 19–29. [CrossRef]
156. Gebreab, S.A.; Salah, K.; Jayaraman, R.; ur Rehman, M.H.; Ellaham, S. LLM-Based Framework for Administrative Task Automation in Healthcare. In Proceedings of the 2024 12th International Symposium on Digital Forensics and Security (ISDFS), San Antonio, TX, USA, 29–30 April 2024; pp. 1–7.
157. Pashangpour, S.; Nejat, G. The Future of Intelligent Healthcare: A Systematic Analysis and Discussion on the Integration and Impact of Robots Using Large Language Models for Healthcare. *Robotics* **2024**, *13*, 112. [CrossRef]
158. Pears, M.; Konstantinidis, S. The Impact of Aligning Artificial Intelligence Large Language Models With Bloom's Taxonomy in Healthcare Education. *Adv. Bus. Inf. Syst. Anal. Book Ser.* **2024**. [CrossRef]
159. Jiang, X.; Yan, L.; Vavekanand, R.; Hu, M. Large Language Models in Healthcare Current Development and Future Directions. *Preprints* **2024**. [CrossRef]
160. Liu, J.; Wang, C.; Liu, S. Applications of Large Language Models in Clinical Practice: Path, Challenges, and Future Perspectives. *OSF Prepr.* **2024**. [CrossRef]
161. Tustumi, F.; Andreollo, N.A.; de Aguilar-Nascimento, J.E. Future of the language models in healthcare: The role of chatgpt. *ABCD* **2023**, *36*, e1727. [CrossRef]
162. Wade, E.C.; Stirman, S.W.; Ungar, L.H.; Boland, C.L.; Schwartz, H.A.; Yaden, D.B.; Sedoc, J.; DeRubeis, R.J.; Willer, R.; Eichstaedt, J.C. Large language models could change the future of behavioral healthcare: A proposal for responsible development and evaluation. *npj Ment. Health Res.* **2024**, *3*, 12. [CrossRef]
163. Sharaf, S.; Anoop, V. An Analysis on Large Language Models in Healthcare: A Case Study of BioBERT. *arXiv* **2023**. [CrossRef]
164. Xia, X.; Jin, Z.; Aiello, M.; Zhang, D.; Liang, G.; Hu, X. Software Service Engineering in the Era of Large Language Models. In Proceedings of the 2024 IEEE International Conference on Software Services Engineering (SSE), IEEE Computer Society, Shenzhen, China, 7–13 July 2024; p. xxiii.
165. Barham, H.; Fasha, M. Towards LLMCI—Multimodal AI for LLM-Vision UI Operation. *Preprint* **2024**. [CrossRef]
166. Chen, T. Challenges and Opportunities in Integrating LLMs into Continuous Integration/Continuous Deployment (CI/CD) Pipelines. In Proceedings of the 2024 5th International Seminar on Artificial Intelligence, Networking and Information Technology (AINIT), Nanjing, China, 29–31 March 2024. [CrossRef]
167. Kulkarni, A.; Shivananda, A.; Kulkarni, A.; Gudivada, D. LLMs for Enterprise and LLMOps. In *Applied Generative AI for Beginners: Practical Knowledge on Diffusion Models, ChatGPT, and Other LLMs*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 117–154.
168. Schillaci, Z. On-Site Deployment of LLMs. In *Large Language Models in Cybersecurity: Threats, Exposure and Mitigation*; Springer Nature Switzerland: Cham, Switzerland, 2024; pp. 205–211.

169. Asmitha, M.; Danda, A.; Bysani, H.; Singh, R.P.; Kanchan, S. Automation of Text Summarization Using Hugging Face NLP. In Proceedings of the 2024 5th International Conference for Emerging Technology (INCET), Belgaum, India, 24–26 May 2024; pp. 1–7.
170. Wang, J.; Chen, N.; Sun, Q.; Huang, W.; Wang, C.; Gao, M. HugNLP: A Unified and Comprehensive Library for Natural Language Processing. In Proceedings of the 32nd ACM International Conference on Information and Knowledge Management, Birmingham, UK, 21–25 October 2023; pp. 5111–5116.
171. Jacob, T.P.; Bizotto, B.L.S.; Sathiyanarayanan, M. Constructing the ChatGPT for PDF Files with Langchain–AI. In Proceedings of the 2024 International Conference on Inventive Computation Technologies (ICICT), Lalitpur, Nepal, 24–26 April 2024; pp. 835–839.
172. Micheal, A.A.; Prasanth, A.; Aswin, T.; Krisha, B. Advancing Educational Accessibility: The LangChain LLM Chatbot's Impact on Multimedia Syllabus-Based Learning. *Preprint* **2024**. [CrossRef]
173. Rahman, M.A. A Survey on Security and Privacy of Multimodal LLMs-Connected Healthcare Perspective. In Proceedings of the 2023 IEEE Globecom Workshops (GC Wkshps), Kuala Lumpur, Malaysia, 4–8 December 2023; pp. 1807–1812.
174. Topsakal, O.; Akinci, T.C. Creating large language model applications utilizing langchain: A primer on developing llm apps fast. In Proceedings of the International Conference on Applied Engineering and Natural Sciences, Konya, Turkey, 10–12 July 2023; Volume 1, pp. 1050–1056.
175. Huh, J.; Park, H.J.; Ye, J.C. Breast ultrasound report generation using LangChain. *arXiv* **2023**, arXiv:2312.03013.
176. Burgan, C.; Kowalski, J.; Liao, W. Developing a Retrieval Augmented Generation (RAG) Chatbot App Using Adaptive Large Language Models (LLM) and LangChain Framework. *Proc. West Va. Acad. Sci.* **2024**, *96*. [CrossRef]
177. Ananthajothi, K.; David, J.; Kavin, A. Cardiovascular Disease Prediction Using Langchain. In Proceedings of the 2024 International Conference on Advances in Computing, Communication and Applied Informatics (ACCAI), Chennai, India, 9–10 May 2024; pp. 1–6.
178. Asyrofi, R.; Dewi, M.R.; Lutfhi, M.I.; Wibowo, P. Systematic Literature Review Langchain Proposed. In Proceedings of the 2023 International Electronics Symposium (IES), Denpasar, Indonesia, 8–10 August 2023; pp. 533–537.
179. Jacobs, S.A.; Tanaka, M.; Zhang, C.; Zhang, M.; Song, L.; Rajbhandari, S.; He, Y. Deepspeed ulysses: System optimizations for enabling training of extreme long sequence transformer models. *arXiv* **2023**, arXiv:2309.14509.
180. Holmes, C.; Tanaka, M.; Wyatt, M.; Awan, A.A.; Rasley, J.; Rajbhandari, S.; Aminabadi, R.Y.; Qin, H.; Bakhtiari, A.; Kurilenko, L.; et al. Deepspeed-fastgen: High-throughput text generation for llms via mii and deepspeed-inference. *arXiv* **2024**, arXiv:2401.08671.
181. Hanindhito, B.; Patel, B.; John, L.K. Bandwidth Characterization of DeepSpeed on Distributed Large Language Model Training. In Proceedings of the 2024 IEEE International Symposium on Performance Analysis of Systems and Software (ISPASS), Indianapolis, Indiana, 5–7 May 2024; pp. 241–256.
182. Cloud, G. Dialogflow CX Documentation. Available online: https://cloud.google.com/dialogflow/cx/docs (accessed on 7 September 2024).
183. Barus, S.P.; Surijati, E. Chatbot with dialogflow for FAQ services in Matana university library. *Int. J. Inform. Comput.* **2022**, *3*, 51–62. [CrossRef]
184. Dall'Acqua, A.; Tamburini, F. Implementing a pragmatically adequate chatbot in DialogFlow CX. In Proceedings of the Ceur Workshop Proceedings, CEUR-WS, Naples, Italy, 4–5 October 2021; Volume 3033, pp. 1–8.
185. Sung, M.C. Development of Prospective Teachers' Adaptive Expertise in AI Chatbots: Comparative Analyses of Google Dialogflow ES and CX. *Multimed.-Assist. Lang. Learn.* **2022**, *25*, 132–151.
186. Microsoft. Azure OpenAI Service. Available online: https://azure.microsoft.com/en-us/products/ai-services/openai-service (accessed on 7 September 2024).
187. Bisson, S.; Branscombe, M.; Hoder, C.; Raman, A. *Azure AI Services at Scale for Cloud, Mobile, and Edge*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2022.
188. LaNeve, J.; Naik, K. Building and Deploying LLM Applications with Apache Airflow. 2023. Available online: https://airflowsummit.org/slides/2023/Building_and_deploying_LLM_applications_with_Apache_Airflow___Airflow_Summit_2023.pdf (accessed on 7 September 2024).
189. Dureja, P. Harnessing Apache Airflow Operators for Enhanced Workflow Automation. *J. Artif. Intell. Cloud Comput.* **2023**, *2*, 1–3. [CrossRef]
190. Amazon Web Services (AWS). Deploy Large Language Models on AWS Inferentia2 Using Large Model Inference Containers. Available online: https://aws.amazon.com/blogs/machine-learning/deploy-large-language-models-on-aws-inferentia2-using-large-model-inference-containers/ (accessed on 7 September 2024).
191. Fregly, C.; Barth, A.; Eigenbrode, S. *Generative AI on AWS: Building Context-Aware Multimodal Reasoning Applications*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2023.

192. Katti, J.; Agarwal, J.; Bharata, S.; Shinde, S.; Mane, S.; Biradar, V. University admission prediction using Google Vertex AI. In Proceedings of the 2022 First International Conference on Artificial Intelligence Trends and Pattern Recognition (ICAITPR), Hyderabad, India, 10–12 March 2022; pp. 1–5.
193. Wang, H.; Yang, J.; Liang, G.; Lee, Y.; Cao, Z. Analyzing the Usability, Performance, and Cost-Efficiency of Deploying ML Models on BigQuery ML and Vertex AI in Google Cloud. In Proceedings of the 2024 8th International Conference on Cloud and Big Data Computing, Oxford, UK, 15–17 August 2024; pp. 15–25.
194. Jabbari, R.; bin Ali, N.; Petersen, K.; Tanveer, B. What is DevOps? A systematic mapping study on definitions and practices. In Proceedings of the Scientific Workshop Proceedings of XP2016, Scotland, UK, 24 May 2016; pp. 1–11.
195. Kadaskar, H.R. Unleashing the Power of Devops in Software Development. *Int. J. Sci. Res. Mod. Sci. Technol.* **2024**, *3*, 1–7.
196. Azad, N.; Hyrynsalmi, S. Multivocal Literature Review on DevOps Critical Success Factors. In Proceedings of the EASE '24: Proceedings of the 28th International Conference on Evaluation and Assessment in Software Engineering, Salerno, Italy, 18–21 June 2024. [CrossRef]
197. Karunarathne, M.; Wijayanayake, W.J.I.; Prasadika, A.P.K.J. DevOps Adoption in Software Development Organizations: A Systematic Literature Review. In Proceedings of the 2024 4th International Conference on Advanced Research in Computing (ICARC), Belihuloya, Sri Lanka, 21–24 February 2024; pp. 282–287. [CrossRef]
198. GeeksforGeeks. DevOps Lifecycle. 2023. Available online: https://www.geeksforgeeks.org/devops-lifecycle/ (accessed on 12 October 2024).
199. Kusumadewi, R.; Adrian, R. Performance Analysis of Devops Practice Implementation of CI/CD Using Jenkins. *J. Comput. Sci. Inf. Technol.* **2023**, *15*, 90–95. [CrossRef]
200. Moutsatsos, I.; Hossain, I.; Agarinis, C.; Harbinski, F.; Abraham, Y.; Dobler, L.; Zhang, X.; Wilson, C.D.; Jenkins, J.L.; Holway, N.; et al. Jenkins-CI, an Open-Source Continuous Integration System, as a Scientific Data and Image-Processing Platform. *J. Biomol. Screen.* **2017**, *22*, 238–249. [CrossRef]
201. Rai, P.; Madhurima; Dhir, S.; Madhulika; Garg, A. A prologue of JENKINS with comparative scrutiny of various software integration tools. In Proceedings of the 2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom), New Delhi, India, 11–13 March 2015; pp. 201–205.
202. Kanchana, A.; Chandrashekar Murthy, B.N. Automated Development and Testing of ECUs in Automotive Industry with Jenkins. In Proceedings of the 2020 IEEE International Conference on Electronics, Computing and Communication Technologies (CONECCT), Online, 2–4 July 2020; pp. 1–5.
203. Arefeen, M.S.; Schiller, M. Continuous integration using GitLab. *Undergrad. Res. Nat. Clin. Sci. Technol. J.* **2019**, *3*, 1–6. [CrossRef]
204. Sharif, M.; Janto, S.; Lueckemeyer, G. Coaas: Continuous integration and delivery framework for hpc using gitlab-runner. In Proceedings of the 2020 4th International Conference on Big Data and Internet of Things, Fuzhou, China, 12–14 June 2020; pp. 54–58.
205. Reddy, A.K.; Alluri, V.R.R.; Thota, S.; Ravi, C.S.; Bonam, V.S.M. DevSecOps: Integrating Security into the DevOps Pipeline for Cloud-Native Applications. *J. Artif. Intell. Res. Appl.* **2021**, *1*, 89–114.
206. Gallaba, K.; Lamothe, M.; McIntosh, S. Lessons from eight years of operational data from a continuous integration service: An exploratory case study of circleci. In Proceedings of the 44th International Conference on Software Engineering, Pittsburgh, PA, USA, 25–27 May 2022; pp. 1330–1342.
207. Hung, P.D.; Giang, D.T. Continuous Integration for Android Application Development and Training. In Proceedings of the ICEMT '19: Proceedings of the 3rd International Conference on Education and Multimedia Technology, Nagoya, Japan, 22–25 July 2019. [CrossRef]
208. Sochat, V. Containershare: Open Source Registry to build, test, deploy with CircleCI. *J. Open Source Softw.* **2018**, *3*, 878. [CrossRef]
209. Krief, M. *Learning DevOps: The Complete Guide to Accelerate Collaboration with Jenkins, Kubernetes, Terraform and Azure DevOps*; Packt Publishing Ltd.: Birmingham, UK, 2019.
210. Jackson, S. Setting Up Azure DevOps. In *Accelerating Unity Through Automation: Power Up Your Unity Workflow by Offloading Intensive Tasks*; Springer: Berlin/Heidelberg, Germany, 2023; pp. 193–243.
211. Borra, P. Maximizing Efficiency and Collaboration with Microsoft Azure DevOps. *Planning* **2024**, *4*, 556–562.
212. Mustyala, A. CI/CD Pipelines in Kubernetes: Accelerating Software Development and Deployment. *Eph-Int. J. Sci. Eng.* **2022**, *8*, 1–11.
213. Shevchuk, R.; Karpinski, M.; Kasianchuk, M.; Yakymenko, I.; Melnyk, A.; Tykhyi, R. Software for Improve the Security of Kubernetes-based CI/CD Pipeline. In Proceedings of the 2023 13th International Conference on Advanced Computer Information Technologies (ACIT), Wrocław, Poland, 21–23 September 2023; pp. 420–425.
214. Schmeling, B.; Dargatz, M. Kubernetes-Native Pipelines. In *Kubernetes Native Development: Develop, Build, Deploy, and Run Applications on Kubernetes*; Springer: Berlin/Heidelberg, Germany, 2022; pp. 243–302.

215. Benedetti, P.; Coviello, G.; Rao, K.; Chakradhar, S. Scale Up while Scaling Out Microservices in Video Analytics Pipelines. In Proceedings of the 2023 IEEE 16th International Symposium on Embedded Multicore/Many-core Systems-on-Chip (MCSoC), Singapore, 18–21 December 2023; pp. 584–591.
216. Bahaweres, R.B.; Zulfikar, A.; Hermadi, I.; Suroso, A.I.; Arkeman, Y. Docker and Kubernetes Pipeline for DevOps Software Defect Prediction with MLOps Approach. In Proceedings of the 2022 2nd International Seminar on Machine Learning, Optimization, and Data Science (ISMODE), Jakarta, Indonesia, 22–23 December 2022; pp. 248–253.
217. Atkinson, B.; Edwards, D. *Generic Pipelines Using Docker: The DevOps Guide to Building Reusable, Platform Agnostic CI/CD Frameworks*; Apress: Berkeley, CA, USA, 2018.
218. Fernández González, D.; Rodríguez Lera, F.J.; Esteban, G.; Fernández Llamas, C. Secdocker: Hardening the continuous integration workflow: Wrapping the container layer. *SN Comput. Sci.* **2022**, *3*, 1–13. [CrossRef]
219. Arapidis, C. *Sonar Code Quality Testing Essentials*; Packt Publishing: Birmingham, UK, 2012.
220. Andrade, M.J. White-Box Testing Automation With SonarQube: Continuous Integration, Code Review, Security, and Vendor Branches. In *Code Generation, Analysis Tools, and Testing for Quality*; IGI Global: Hershey, PA, USA, 2019; pp. 64–88.
221. Artefact Engineering and Data Science. Why You Need LLMOps. 2024. Available online: https://medium.com/artefact-engineering-and-data-science/why-you-need-llmops-48c0925827de (accessed on 25 December 2024).
222. Puget Systems Team. LLM Inference Consumer GPU Performance. 2023. Available online: https://www.pugetsystems.com/labs/articles/llm-inference-consumer-gpu-performance/ (accessed on 21 December 2024).
223. Matics Analytics Team. LLMOps: Scaling LLM Deployment with 100% Throughput Improvement. 2023. Available online: https://www.maticsanalytics.com/post/llmops-scaling-llm-deployment-with-~100-throughput-improvement (accessed on 6 October 2024).
224. Jeong, C. A study on the implementation of generative ai services using an enterprise data-based llm application architecture. *arXiv* **2023**, arXiv:2309.01105. [CrossRef]
225. Kim, T.; Wang, Y.; Chaturvedi, V.; Gupta, L.; Kim, S.; Kwon, Y.; Ha, S. LLMem: Estimating GPU Memory Usage for Fine-Tuning Pre-Trained LLMs. *arXiv* **2024**, arXiv:2404.10933.
226. DataCamp Team. TPU vs GPU: What's the Difference for AI? 2023. Available online: https://www.datacamp.com/blog/tpu-vs-gpu-ai (accessed on 6 October 2024).
227. Incubity by Ambilio Team. GPU vs TPU for LLM Training: A Comprehensive Analysis. 2024. Available online: https://incubity.ambilio.com/gpu-vs-tpu-for-llm-training-a-comprehensive-analysis/ (accessed on 6 October 2024).
228. Google Cloud. Google Cloud TPU. Available online: https://cloud.google.com/tpu?hl=en (accessed on 6 October 2024).
229. Wu, Y.E.; Wu, H.I.; Chin, K.C.; Yang, Y.C.; Tsay, R.S. Accelerate Large Language Model Inference on Edge TPU with OpenVX framework. In Proceedings of the 2024 IEEE 6th International Conference on AI Circuits and Systems (AICAS), Abu Dhabi, United Arab Emirates, 22–25 April 2024; pp. 502–506.
230. Brakel, F.; Odyurt, U.; Varbanescu, A.L. Model Parallelism on Distributed Infrastructure: A Literature Review from Theory to LLM Case-Studies. *arXiv* **2024**, arXiv:2403.03699.
231. Carrión, D.S.; Prohaska, V. Exploration of TPUs for AI Applications. *arXiv* **2023**, arXiv:2309.08918.
232. Run:ai Team. Parallelism Strategies for Distributed Training. 2023. Available online: https://www.run.ai/blog/parallelism-strategies-for-distributed-training (accessed on 6 October 2024).
233. Kim, S.; Moon, S.; Tabrizi, R.; Lee, N.; Mahoney, M.W.; Keutzer, K.; Gholami, A. An LLM compiler for parallel function calling. *arXiv* **2023**, arXiv:2312.04511.
234. Singh, S.; Karatzas, A.; Fore, M.; Anagnostopoulos, I.; Stamoulis, D. An LLM-Tool Compiler for Fused Parallel Function Calling. *arXiv* **2024**, arXiv:2405.17438.
235. Low, Y.; Gonzalez, J.E.; Kyrola, A.; Bickson, D.; Guestrin, C.E.; Hellerstein, J. Graphlab: A new framework for parallel machine learning. *arXiv* **2014**, arXiv:1408.2041.
236. Bender, E.M.; Gebru, T.; McMillan-Major, A.; Shmitchell, S. On the dangers of stochastic parrots: Can language models be too big? In Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event, 3–10 March 2021; pp. 610–623.
237. Ni, C.; Wu, J.; Wang, H.; Lu, W.; Zhang, C. Enhancing cloud-based large language model processing with elasticsearch and transformer models. In Proceedings of the International Conference on Image, Signal Processing, and Pattern Recognition (ISPP 2024), Guangzhou, China, 1–3 March 2024; Volume 13180, pp. 1648–1654.
238. Tamkin, A.; Brundage, M.; Clark, J.; Ganguli, D. Understanding the capabilities, limitations, and societal impact of large language models. *arXiv* **2021**, arXiv:2102.02503.
239. Luu, H.; Pumperla, M.; Zhang, Z. The Future of MLOps. In *MLOps with Ray: Best Practices and Strategies for Adopting Machine Learning Operations*; Springer: Berlin/Heidelberg, Germany, 2024; pp. 305–327.
240. Liu, Y.; Xu, Y.; Song, R. Transforming User Experience (UX) through Artificial Intelligence (AI) in interactive media design. *Eng. Sci. Technol. J.* **2024**, *5*, 2273–2283. [CrossRef]

241. Jin, H.; Zhang, Y.; Meng, D.; Wang, J.; Tan, J. A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods. *arXiv* **2024**, arXiv:2403.02901.
242. Alammar, J.; Grootendorst, M. *Hands-On Large Language Models*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2024.
243. Jones, B.; Dixon, G. Boosting Textual Understanding in LLMs with Context-Aware Flexible Length Tokenization. OSF Preprints. Available online: https://osf.io/preprints/osf/9gnjt (accessed on 24 December 2024).
244. Mumuni, A.; Mumuni, F. Automated data processing and feature engineering for deep learning and big data applications: A survey. *J. Inf. Intell.* **2024**, *in press*. [CrossRef]
245. Digital Alpha Team. LLMOps Unveiled: Your Step-by-Step Guide to Building Production-Ready LLM Applications. 2023. Available online: https://www.digital-alpha.com/llmops-unveiled-your-step-by-step-guide-to-building-production-ready-llm-applications/ (accessed on 6 October 2024).
246. CHen, Z.; Cao, L.; Madden, S.; Fan, J.; Tang, N.; Gu, Z.; Shang, Z.; Liu, C.; Cafarella, M.; Kraska, T. Seed: Simple, efficient, and effective data management via large language models. *arXiv* **2023**, arXiv:2310.00749.
247. Pansara, R.R. NoSQL Databases and Master Data Management: Revolutionizing Data Storage and Retrieval. *Int. Numer. J. Mach. Learn. Robot.* **2020**, *4*, 1–11.
248. Gupta, I.; Singh, A.K.; Lee, C.N.; Buyya, R. Secure data storage and sharing techniques for data protection in cloud environments: A systematic review, analysis, and future directions. *IEEE Access* **2022**, *10*, 71247–71277. [CrossRef]
249. Nambiar, A.; Mundra, D. An overview of data warehouse and data lake in modern enterprise data management. *Big Data Cogn. Comput.* **2022**, *6*, 132. [CrossRef]
250. Kehrer, K.; Kaiser, C. *Machine Learning Upgrade: A Data Scientist's Guide to MLOps, LLMs, and ML Infrastructure*; John Wiley & Sons: Hoboken, NJ, USA, 2024.
251. Wang, X.; Kim, H.; Rahman, S.; Mitra, K.; Miao, Z. Human-LLM collaborative annotation through effective verification of LLM labels. In Proceedings of the CHI Conference on Human Factors in Computing Systems, Honolulu, HI, USA, 11–16 May 2024; pp. 1–21.
252. Kim, H.; Mitra, K.; Chen, R.L.; Rahman, S.; Zhang, D. Meganno+: A human-llm collaborative annotation system. *arXiv* **2024**, arXiv:2402.18050.
253. Tan, Z.; Beigi, A.; Wang, S.; Guo, R.; Bhattacharjee, A.; Jiang, B.; Karami, M.; Li, J.; Cheng, L.; Liu, H. Large language models for data annotation: A survey. *arXiv* **2024**, arXiv:2402.13446.
254. Sutharsan, M. Smart analysis of automated and semi-automated approaches to data annotation for machine learning. *ICTACT J. Data Sci. Mach. Learn.* **2023**, *4*, 457–460.
255. Pustejovsky, J.; Stubbs, A. *Natural Language Annotation for Machine Learning: A Guide to Corpus-Building for Applications*; O'Reilly Media, Inc.: Sebastopol, CA, USA, 2012.
256. Van Atteveldt, W.; Van der Velden, M.A.; Boukes, M. The validity of sentiment analysis: Comparing manual annotation, crowd-coding, dictionary approaches, and machine learning algorithms. *Commun. Methods Meas.* **2021**, *15*, 121–140. [CrossRef]
257. Pathania, N. *Learning Continuous Integration with Jenkins*; Packt Publishing Ltd.: Birmingham, UK, 2016.
258. Zhao, Z.; Chen, Y.; Bangash, A.A.; Adams, B.; Hassan, A.E. An empirical study of challenges in machine learning asset management. *Empir. Softw. Eng.* **2024**, *29*, 98. [CrossRef]
259. Safri, H.; Papadimitriou, G.; Deelman, E. Dynamic Tracking, MLOps, and Workflow Integration: Enabling Transparent Reproducibility in Machine Learning. In Proceedings of the 2024 IEEE 20th International Conference on e-Science (e-Science), Osaka, Japan, 16–20 September 2024; pp. 1–10.
260. Semmelrock, H.; Ross-Hellauer, T.; Kopeinik, S.; Theiler, D.; Haberl, A.; Thalmann, S.; Kowald, D. Reproducibility in Machine Learning-based Research: Overview, Barriers and Drivers. *arXiv* **2024**, arXiv:2406.14325.
261. Wu, Z. Sky Computing with Intercloud Brokers. Ph.D. Thesis, University of California, Berkeley, CA, USA, 2024.
262. Sikeridis, D.; Papapanagiotou, I.; Rimal, B.P.; Devetsikiotis, M. A Comparative taxonomy and survey of public cloud infrastructure vendors. *arXiv* **2017**, arXiv:1710.01476.
263. Estrin, E. *Cloud Security Handbook: Find Out How to Effectively Secure Cloud Environments Using AWS, Azure, and GCP*; Packt Publishing Ltd.: Birmingham, UK, 2022.
264. Esas, O. Design Patterns and Anti-Patterns in Microservices Architecture: A Classification Proposal and Study on Open Source Projects. Master's Thesis, Politecnico di Milano, Milano, Italy, 2020. Available online: https://www.politesi.polimi.it/handle/10589/186745 (accessed on 24 December 2024).
265. Huang, Y.; Wan, L.J.; Ye, H.; Jha, M.; Wang, J.; Li, Y.; Zhang, X.; Chen, D. New Solutions on LLM Acceleration, Optimization, and Application. *arXiv* **2024**, arXiv:2406.10903.
266. Isaev, M.; McDonald, N.; Vuduc, R. Scaling infrastructure to support multi-trillion parameter LLM training. In Proceedings of the Architecture and System Support for Transformer Models (ASSYST@ ISCA 2023), Oralndo, FL, USA, 17 June 2023.
267. Srivatsa, K.G. Leveraging Large Language Models for Generating Infrastructure as Code: Open and Closed Source Models and Approaches. Ph.D. Thesis, International Institute of Information Technology Hyderabad, Hyderabad, India, 2024.

268. David, R.B. Kubernetes Auto-Scaling: YoYo Attack Vulnerability and Mitigation. Master's Thesis, Reichman University, Herzliya, Israel, 2021.
269. Ekanayaka, E.M.I.M.; Thathsarani, J.K.K.H.; Karunanayaka, D.S.; Kuruwitaarachchi, N.; Skandakumar, N. Enhancing DevOps Infrastructure for Efficient Management of Microservice Applications. In Proceedings of the 2023 IEEE International Conference on e-Business Engineering (ICEBE), Sydney, Australia, 4–6 November 2023; pp. 63–68.
270. Mohajeri, M.A. Leveraging Large Language Model for Enhanced Business Analytics on AWS. Master's Thesis, Centria University of Applied Sciences, Kokkola, Finland, 2024. Available online: https://www.theseus.fi/handle/10024/859982 (accessed on 12 December 2024).
271. Li, H.; Wang, S.X.; Shang, F.; Niu, K.; Song, R. Applications of large language models in cloud computing: An empirical study using real-world data. *Int. J. Innov. Res. Comput. Sci. Technol.* **2024**, *12*, 59–69. [CrossRef]
272. Zhou, X.; Zhao, X.; Li, G. LLM-Enhanced Data Management. *arXiv* **2024**, arXiv:2402.02643.
273. Wang, Z.; Zhong, W.; Wang, Y.; Zhu, Q.; Mi, F.; Wang, B.; Shang, L.; Jiang, X.; Liu, Q. Data management for large language models: A survey. *arXiv* **2023**, arXiv:2312.01700.
274. Webber, E.; Olgiati, A. *Pretrain Vision and Large Language Models in Python: End-to-End Techniques for Building and Deploying Foundation Models on AWS*; Packt Publishing Ltd.: Birmingham, UK, 2023.
275. Capizzi, A.; Distefano, S.; Mazzara, M. From devops to devdataops: Data management in devops processes. In Proceedings of the Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment: Second International Workshop, DEVOPS 2019, Château de Villebrumier, France, 6–8 May 2019; Revised Selected Papers 2; Springer: Berlin/Heidelberg, Germany, 2020; pp. 52–62.
276. Chawla, H.; Khattar, P.; Chawla, H.; Khattar, P. Building Blocks of Data Analytics. In *Data Lake Analytics on Microsoft Azure*; Apress: Berkeley, CA, USA, 2020; pp. 11–25.
277. Chan, Y.C.; Pu, G.; Shanker, A.; Suresh, P.; Jenks, P.; Heyer, J.; Denton, S. Balancing Cost and Effectiveness of Synthetic Data Generation Strategies for LLMs. *arXiv* **2024**, arXiv:2409.19759.
278. Liu, Y.; Zhang, H.; Miao, Y.; Le, V.H.; Li, Z. OptLLM: Optimal Assignment of Queries to Large Language Models. *arXiv* **2024**, arXiv:2405.15130.
279. Kaswan, S.; Goyal, P.; Khirasaria, V.; Yugal, L.; Amita, E. Current Trends and Challenges of Cloud Computing and Emerging Technological Utilization in Technical Education. In Proceedings of the 2024 5th International Conference on Electronics and Sustainable Communication Systems (ICESC), Coimbatore, India, 7–9 August 2024; pp. 629–634.
280. Patel, P.; Choukse, E.; Zhang, C.; Goiri, Í.; Warrier, B.; Mahalingam, N.; Bianchini, R. Characterizing Power Management Opportunities for LLMs in the Cloud. In Proceedings of the 29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, La Jolla, CA, USA, 27 April–1 May 2024; Volume 3, pp. 207–222.
281. Thati, B.; Shyam, K.M.; Sindhura, S.; Pulletikurthy, D.; Chowdary, N.S. Continuous Deployment in Action: Developing a Cloud-Based Image Matching Game. *Int. J. Innov. Technol. Interdiscip. Sci.* **2024**, *7*, 68–79.
282. Rangnau, T.; Buijtenen, R.V.; Fransen, F.; Turkmen, F. Continuous security testing: A case study on integrating dynamic security testing tools in ci/cd pipelines. In Proceedings of the 2020 IEEE 24th International Enterprise Distributed Object Computing Conference (EDOC), Eindhoven, The Netherlands, 5–8 October 2020; pp. 145–154.
283. Choudhary, S. Kubernetes-Based Architecture For an On-premises Machine Learning Platform. Master's Thesis, Aalto University, Espoo, Finland, 2021.
284. Wilkins, G.; Keshav, S.; Mortier, R. Hybrid Heterogeneous Clusters Can Lower the Energy Consumption of LLM Inference Workloads. In Proceedings of the 15th ACM International Conference on Future and Sustainable Energy Systems, Singapore, 4–7 June 2024; pp. 506–513.
285. Mienye, I.D.; Swart, T.G.; Obaido, G. XtremeLLMs: Towards Extremely Large Language Models. *Preprints* **2024**. Available online: https://www.preprints.org/manuscript/202408.1483/v1 (accessed on 14 December 2024).
286. Yu, S.; Fang, C.; Ling, Y.; Wu, C.; Chen, Z. LLM for Test Script Generation and Migration: Challenges, Capabilities, and Opportunities. *arXiv* **2023**. [CrossRef]