Scaling Machine Learning Model Training with CI/CD Pipelines in Cloud Environments

Swamy Prasadarao Velaga

Sr. Program Automation Architect, Department of Information Technology

Abstract:
As machine learning (ML) continues to advance, the need for scalable, efficient, and reliable model training has become critical. Traditional approaches to ML model training often struggle to meet these demands, prompting the integration of Continuous Integration and Continuous Deployment (CI/CD) practices with cloud environments. This survey paper explores the intersection of CI/CD pipelines and cloud-based solutions in scaling ML model training. We provide a comprehensive review of the current state of CI/CD practices tailored for ML workflows, examine the benefits and offerings of cloud environments, and identify best practices, tools, and frameworks that facilitate this integration. Additionally, we address the challenges associated with resource management, data handling, distributed training, model versioning, and security. By leveraging cloud-native tools and adhering to best practices, organizations can optimize their ML workflows, ensuring efficient and consistent model updates. Furthermore, we highlight future research directions, including advanced resource management techniques, federated learning, AI-driven automation, standardization, enhanced security frameworks, explainability, fairness, and sustainable AI practices. This paper aims to serve as a valuable resource for researchers, practitioners, and organizations seeking to optimize their ML workflows through the effective implementation of CI/CD pipelines in cloud environments, ultimately leading to more robust, reliable, and ethical AI systems.

Keywords: Continuous Deployment, AI Systems, Machine Learning Models, Cloud Environments

1. Introduction

In recent years, the field of machine learning (ML) has seen unprecedented growth, driven by advancements in algorithms, increased computational power, and the availability of vast amounts of data [1]. As organizations strive to leverage ML for various applications, the need to train and deploy models efficiently and at scale has become paramount. Traditional ML workflows, often characterized by manual and ad-hoc processes, struggle to meet the demands of modern, large-scale applications. This has led to the adoption of Continuous Integration and Continuous Deployment (CI/CD) practices, which have revolutionized software development by enabling rapid, reliable, and repeatable deployment processes [1].

The convergence of CI/CD methodologies with ML model training and deployment is a natural progression, aimed at addressing the challenges associated with scaling ML workflows [2]. CI/CD pipelines facilitate automated testing, integration, and deployment, ensuring that models are consistently updated and maintained. When combined with the scalability and flexibility offered by cloud environments, CI/CD pipelines become a powerful tool for managing the entire ML lifecycle, from data ingestion and model training to deployment and monitoring. The role of CI/CD pipeline in cloud environment is presented into Figure 1.

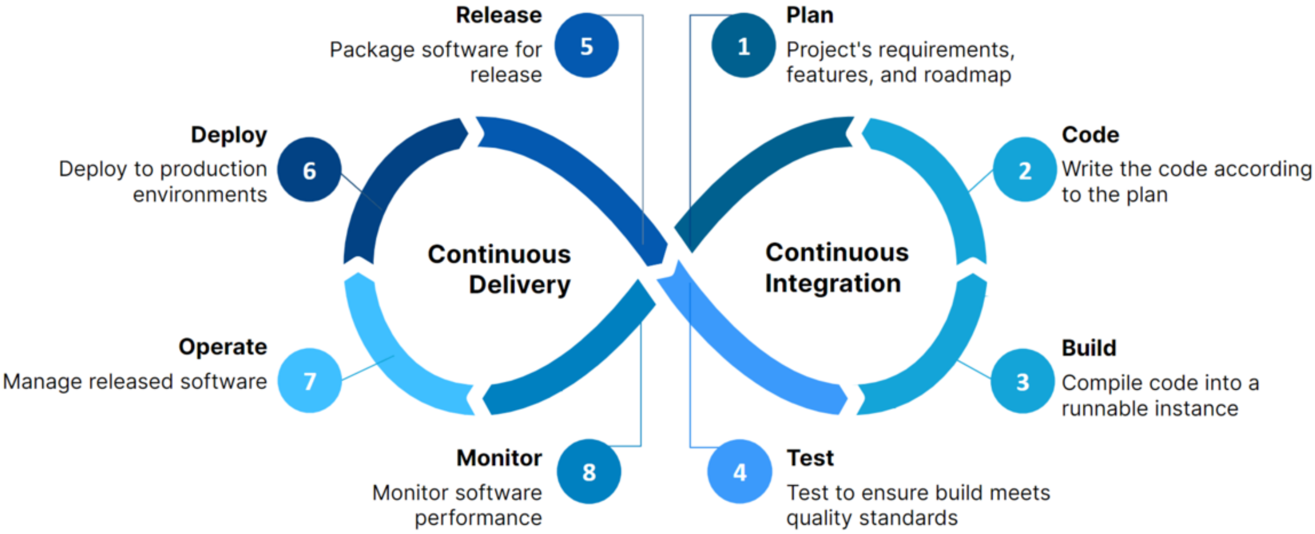

Figure 1: The Role of CI/CD pipeline in Cloud Environment[1]

This survey aims to explore the intersection of CI/CD pipelines and cloud environments in the context of scaling ML model training. The primary objectives are:
1. To review the current state of CI/CD practices tailored for ML workflows.
2. To examine the role of cloud environments in enhancing the scalability of ML model training.
3. To identify best practices, tools, and frameworks that facilitate the integration of CI/CD pipelines with cloud-based ML training.
4. To discuss the challenges and solutions associated with scaling ML model training in cloud environments.
5. To explore future trends and advancements in this rapidly evolving field.

Scope
The focus of this survey is on the use of CI/CD pipelines to scale ML model training specifically within cloud environments [3]. This includes an examination of the essential components of CI/CD pipelines adapted for ML, an overview of the major cloud platforms and their capabilities, and a discussion of the challenges unique to this context. By synthesizing insights from recent research and industry practices, this paper aims to provide a comprehensive overview of the strategies and technologies that enable scalable, efficient, and reliable ML model training in the cloud [4].

This survey aims to serve as a valuable resource for researchers, practitioners, and organizations seeking to optimize their ML workflows by leveraging the synergy between CI/CD pipelines and cloud environments.

The paper is structured as follows: Section 2 provides a comprehensive literature review. Section 3 covers CI/CD Pipelines for ML in Cloud, including Automation Tools and Case Studies. Section 4 discusses Challenges and Solutions. Finally, Section 5 concludes with Future Research Directions.

2. Literature Review

This chapter explores the integration of artificial intelligence (AI) and machine learning (ML) into cloud-based communication networks, focusing on the complexities involved in developing, testing, deploying, and securing AI-driven services [5]. Cloud technologies like containers, microservices, and cloud-native applications offer benefits such as application reusability, accelerated time-to-market, and infrastructure optimization, but their management complexities necessitate AI-driven solutions. Microservices in cloud environments

[1] https://www.civo.com/blog/the-role-of-the-ci-cd-pipeline-in-cloud-computing

leverage interprocess communication for flexible service interactions, while virtual machines provide hardware-agnostic deployment capabilities. Network Function Virtualization (NFV) enhances agility by virtualizing network node functionalities as virtual network functions (VNFs), running on generic cloud hardware to optimize resource utilization and scalability in cloud-based communication infrastructures [5].

In recent years, machine learning (ML) and artificial intelligence (AI) have successfully scaled across various sectors like finance, retail, insurance, and energy utilities, enabling tasks such as predicting customer behavior and optimizing financial models [6]. However, their application in healthcare has been limited due to challenges in data quality management, meeting clinical standards, and ensuring regulatory compliance. Existing general-purpose ML platforms often struggle in healthcare settings, which demand rigorous security, privacy, and performance monitoring [6]. This paper introduces Isthmus, a specialized cloud-based platform tailored to overcome these challenges and expedite the deployment of ML/AI solutions in healthcare. Three case studies highlight Isthmus' capabilities: predicting trauma survivability in hospital trauma centers, aggregating data for Social Determinants of Health (SDoH) insights, and leveraging IoT sensor data for real-time predictive analytics [6].

Machine learning (ML) has demonstrated its efficacy in critical web applications like search ranking and is increasingly pivotal across diverse enterprise contexts such as voice recognition, customer support, video conferencing optimization, sysops automation, manufacturing, autonomous vehicles, and financial forecasting [7]. Simultaneously, as the value of data grows and concerns over security and privacy intensify, robust data governance has become imperative in enterprise environments. How will these intersecting trends—ML's expanding role and stringent data governance—affect enterprise operations? What gaps exist in deploying ML in such settings? What technical challenges must the database community address? This paper outlines our vision for integrating ML with database systems and details initial steps toward realizing this vision [7].

This paper outlines recommendations for designing and piloting a curriculum focused on DevOps and Cloud-based Software Development for master's programs in Computer Science and Software Engineering. Central to this approach is the DevOps Body of Knowledge for Software Engineering (DevOpsSE BoK), which defines essential Knowledge Areas and Units necessary for professionals to effectively work as DevOps engineers or application developers [8]. Defining the DevOpsSE BoK establishes a foundation for outlining required competencies and skills, enabling structured and targeted curriculum development. The paper also shares insights from the initial course implementation during the 2018/2019 academic year at the University of Amsterdam [8]. It details the course structure and instructional methodologies employed, such as project-based learning, which enhances students' team-based skills in Agile development processes and fosters knowledge sharing. Additionally, the paper provides an overview of common DevOps definitions, concepts, models, and tools, with a specific focus on cloud-based DevOps tools for software development, deployment, and operations. These tools facilitate continuous development and improvement, essential for modern agile and data-driven enterprises [8].

The XCoLab, initially developed as a single-server architecture by the Center for Collective Intelligence at MIT, serves as an open-source, domain-independent crowdsourcing platform, prominently utilized in initiatives like the Climate CoLab for addressing climate change [9]. The project's current objective is to transition from its existing architecture to a container-based microservices model. This shift promises several benefits: microservices can scale horizontally and independently with minimal configuration adjustments, simply by spawning new container instances. Additionally, adopting a continuous integration (CI) development environment will expedite software updates, mitigating compatibility issues and reducing regression testing time. Furthermore, this overhaul aims to streamline platform deployment, making it infrastructure-agnostic and accessible for new users. Achieving these goals involves selecting suitable technologies, decoupling the 13 platform services, containerizing each service, and rigorously testing their functionalities. Subsequently, configuring CI tools will automate image building, testing, and deployment processes, culminating in a simplified and efficient deployment workflow for the entire containerized platform [9].

In the dynamic field of software engineering, the integration of artificial intelligence (AI) and machine learning (ML) technologies brings forth significant opportunities alongside notable challenges [10]. This paper delves into the essential best practices and hurdles involved in leveraging AI and ML within software engineering processes. Emphasizing the imperative of efficient software engineering practices in the AI and ML era, we underscore critical aspects such as meticulous data management, strategic algorithm selection, seamless model deployment, and ethical considerations. By adeptly addressing these challenges and adhering to best practices, software engineers can effectively harness the potential of AI and ML to develop resilient, scalable software solutions that adeptly meet the evolving demands of modern organizations [10].

The advent of the "big data" era has created a critical need to rapidly and affordably educate numerous data scientists and engineers [11]. It is imperative that they gain practical experience by working on assignments with real-world datasets to develop essential workplace skills. However, facilitating instructors to deliver diverse data science assignments using real-world datasets to a large number of learners, both on-campus and online, poses a significant challenge [11]. To tackle this challenge, we have developed and deployed a novel Cloud-based Lab for Data Science (CLaDS). CLaDS enables learners worldwide to engage in real-world data science projects without the need to handle or distribute large datasets, leveraging version control and continuous integration. This infrastructure allows any instructor to efficiently deliver hands-on data science assignments that utilize substantial datasets to a broad audience at minimal cost. This paper details the design and implementation of CLaDS, showcasing its effectiveness through the deployment of seven major text data assignments in both on-campus and online courses. This demonstrates CLaDS' potential as a versatile platform for delivering diverse data science assignments to a large audience economically and efficiently [11].

Table 1: Summary for The Literature Review

Reference	Model Used	Application	Highlighted Points
[5]	AI and ML technologies in cloud-based communication networks	Development, testing, deployment, and securing AI-driven services	Use of containers, microservices, and NFV in cloud environments; challenges and benefits in deploying AI-driven services.
[6]	Isthmus platform	Healthcare applications: trauma survivability prediction, Social Determinants of Health (SDoH) insights, IoT sensor data analytics	Overcoming challenges in healthcare data management, regulatory compliance, and privacy with specialized ML/AI platform.
[7]	Various ML applications across enterprise contexts	Voice recognition, customer support, video conferencing, sysops automation, manufacturing, autonomous vehicles, financial forecasting	ML's expanding role across diverse sectors; challenges in integrating ML with stringent data governance.
[8]	DevOps and Cloud-based Software Development curricula	Master's programs in Computer Science and Software Engineering	DevOpsSE BoK, curriculum design for DevOps engineers and developers, project-based learning methodologies.
[9]	XCoLab crowdsourcing platform	Crowdsourcing initiatives like Climate CoLab	Transition from single-server to container-based microservices architecture, benefits of scalability and CI/CD integration.
[10]	AI and ML in software engineering processes	Software engineering practices integrating AI and ML	Best practices and challenges in data management, algorithm selection, model deployment, and ethics in AI and ML integration.

[11]	Cloud-based Lab for Data Science (CLaDS)	Education in data science using real-world datasets	Enabling scalable, cost-effective delivery of data science assignments via cloud infrastructure, leveraging version control and continuous integration.

3. CI/CD Pipelines for ML in Cloud

Pipeline Components

To effectively integrate CI/CD principles into ML workflows, it's essential to understand the various components of a CI/CD pipeline specifically tailored for ML in cloud environments [12]. These components include:

1. Data Ingestion and Preprocessing: The pipeline begins with data collection and preprocessing. Automated workflows are used to gather raw data, clean it, and transform it into a format suitable for model training. This step often involves data validation to ensure quality and consistency [12].

2. Model Training: This stage involves the actual training of ML models. In a CI/CD pipeline, this process is automated to trigger whenever new data is available or changes are made to the model code [12]. Cloud environments provide scalable resources, such as distributed computing, to handle large datasets and complex models efficiently.

3. Model Evaluation and Testing: After training, models are evaluated against predefined metrics to ensure they meet performance standards. Automated testing frameworks validate model accuracy, robustness, and fairness. This step includes unit tests, integration tests, and performance tests to detect issues early [13].

4. Model Versioning: Versioning is critical in ML workflows to track changes and manage different iterations of models. CI/CD pipelines automate the versioning process, storing each model version along with its metadata, such as training data, hyperparameters, and evaluation metrics [13].

5. Model Deployment: Once a model passes all evaluation criteria, it is deployed to a production environment. Automated deployment processes ensure that models are consistently and reliably pushed to production. Cloud environments offer various deployment options, such as containerization, serverless functions, and managed ML services [12].

6. Monitoring and Feedback: Post-deployment, the model's performance is continuously monitored. Metrics such as latency, accuracy, and resource usage are tracked to detect deviations from expected behavior. Automated feedback loops can trigger retraining or rollback procedures if performance degrades [13].

Automation Tools

Several tools and frameworks facilitate the implementation of CI/CD pipelines for ML in cloud environments. Key tools include:

1. Kubeflow: An open-source platform that streamlines ML workflows on Kubernetes. Kubeflow integrates with various cloud services, providing tools for data ingestion, model training, deployment, and monitoring [4].

2. MLflow: A platform for managing the ML lifecycle, including experimentation, reproducibility, and deployment. MLflow supports model tracking, packaging, and sharing, making it suitable for CI/CD pipelines in cloud environments [14].

3. Jenkins: A widely-used automation server that supports CI/CD for ML through plugins and integrations with cloud services [15]. Jenkins can orchestrate end-to-end ML workflows, from data preprocessing to model deployment.

4. GitLab CI: A continuous integration service integrated with GitLab. It supports CI/CD for ML by allowing teams to define pipelines as code, automate testing, and deploy models using cloud resources [16].

5. Amazon SageMaker: A managed service by AWS that provides tools for building, training, and deploying ML models. SageMaker integrates seamlessly with other AWS services, facilitating the creation of scalable CI/CD pipelines [17].

Best Practices

Implementing CI/CD pipelines for ML in cloud environments involves several best practices to ensure efficiency, scalability, and reliability:

1. Modular Pipeline Design: Break down the pipeline into modular components that can be independently developed, tested, and maintained. This modularity enhances flexibility and makes it easier to scale and update individual parts of the pipeline [12].

2. Automation of Repetitive Tasks: Automate repetitive and error-prone tasks such as data preprocessing, model training, and deployment. Automation reduces manual intervention, speeds up the workflow, and minimizes the risk of human error.

3. Continuous Testing and Validation: Incorporate comprehensive testing at every stage of the pipeline. Continuous testing ensures that models meet performance standards and helps identify issues early, reducing the risk of deploying faulty models [13].

4. Scalable Infrastructure: Utilize the scalable infrastructure provided by cloud environments to handle varying workloads. Take advantage of cloud-native services, such as auto-scaling and serverless computing, to dynamically allocate resources based on demand.

5. Robust Monitoring and Logging: Implement robust monitoring and logging to track model performance and pipeline health. Use cloud-native monitoring tools to gain insights into resource usage, detect anomalies, and trigger alerts for manual intervention or automated remediation.

6. Security and Compliance: Ensure that the pipeline adheres to security best practices and compliance requirements. Implement data encryption, access controls, and auditing mechanisms to protect sensitive data and maintain regulatory compliance.

Case Studies

1. Spotify: Spotify uses Kubeflow on Google Cloud to manage its ML workflows. By adopting CI/CD practices, Spotify has streamlined its model training and deployment processes, enabling rapid experimentation and continuous delivery of ML models [12].

2. Netflix: Netflix leverages Amazon SageMaker to build, train, and deploy ML models. The integration of CI/CD pipelines with SageMaker has allowed Netflix to scale its ML operations, ensuring that models are consistently updated and optimized for performance [14].

3. Airbnb: Airbnb utilizes MLflow for managing the ML lifecycle, including experimentation and deployment. By integrating MLflow with its CI/CD pipeline, Airbnb has enhanced collaboration among data scientists and improved the reproducibility of its ML models [13].

In conclusion, implementing CI/CD pipelines for ML in cloud environments offers numerous benefits, including scalability, automation, and reliability. By leveraging cloud-native tools and adhering to best practices, organizations can effectively manage the ML lifecycle, from data ingestion to model deployment and monitoring. This integration not only accelerates the development and deployment of ML models but also ensures their continuous improvement and adaptation to changing requirements.

4. Challenges and Solutions

Scalability Issues

Challenges:

1. Resource Management: Efficiently managing computational resources, especially in a cloud environment, is a significant challenge. Training large ML models often requires substantial computational power and memory, which can be costly and difficult to allocate dynamically [16].
2. Data Handling: Handling large volumes of data for training and inference poses challenges in terms of storage, transfer, and processing. Ensuring data availability and consistency across distributed systems can be complex.
3. Distributed Training: Implementing distributed training across multiple nodes introduces issues such as synchronization, communication overhead, and fault tolerance [14]. Ensuring that the training process scales linearly with the addition of resources is non-trivial.

Solutions:
1. Auto-Scaling: Utilize cloud-native auto-scaling features to dynamically allocate and deallocate resources based on workload demands. This helps in optimizing costs while ensuring sufficient resources are available for training.
2. Data Sharding and Partitioning: Implement data sharding and partitioning strategies to distribute data across multiple storage systems. This enables parallel data processing and improves the efficiency of data handling.
3. Distributed Training Frameworks: Use frameworks like TensorFlow, PyTorch, or Horovod that support distributed training out of the box. These frameworks handle synchronization and communication efficiently, reducing the complexity of scaling training processes.

Model Versioning

Challenges:
1. Tracking Changes: Keeping track of changes in model parameters, hyperparameters, and training data over different versions can be cumbersome [12]. Without proper versioning, it's difficult to reproduce results or roll back to previous model versions.
2. Metadata Management: Managing metadata associated with each model version, such as training data, code version, and evaluation metrics, is essential for maintaining an organized workflow.

Solutions:
1. Model Registry: Implement a model registry to track and manage different versions of models. Tools like MLflow, DVC, or custom-built registries can help store metadata and provide version control.
2. Automated Versioning: Integrate automated versioning in the CI/CD pipeline to ensure that every change in the model or its configuration is recorded. This enables easy rollback and reproducibility of experiments.

Security and Compliance

Challenges:
1. Data Privacy: Ensuring the privacy and security of sensitive data used for training ML models is critical. Unauthorized access or data breaches can lead to significant legal and reputational damage [15].

2. Compliance Requirements: Adhering to industry-specific regulations and compliance standards (e.g., GDPR, HIPAA) can be challenging, especially when dealing with cross-border data transfers and storage.

Solutions:
1. Data Encryption: Implement encryption for data at rest and in transit to protect sensitive information. Use cloud-native encryption services to simplify the management of encryption keys.
2. Access Controls: Establish stringent access control policies to restrict access to sensitive data and model artifacts. Use role-based access control (RBAC) and identity and access management (IAM) services provided by cloud platforms [16].
3. Audit and Monitoring: Implement comprehensive audit trails and monitoring to track access and changes to data and models. This helps in maintaining transparency and accountability, essential for compliance.

Model Deployment

Challenges:
1. Deployment Automation: Automating the deployment of ML models while ensuring consistency and reliability can be challenging [17]. Manual interventions often introduce errors and slow down the deployment process.
2. Serving Infrastructure: Managing the infrastructure to serve models at scale requires robust solutions to handle load balancing, scaling, and fault tolerance.

Solutions:
1. Continuous Deployment Tools: Use continuous deployment tools like Jenkins, GitLab CI, or cloud-native services (e.g., AWS CodePipeline, Azure DevOps) to automate the deployment process. These tools can integrate with ML workflows to streamline deployment [16].
2. Containerization: Containerize ML models using Docker and orchestrate deployments with Kubernetes. Containers ensure consistency across environments and make scaling and managing serving infrastructure easier.
3. Managed Services: Leverage managed services like AWS SageMaker, Google AI Platform, or Azure ML for model deployment. These services handle underlying infrastructure complexities, allowing teams to focus on model development and improvement [17].

Scaling ML model training with CI/CD pipelines in cloud environments presents unique challenges, from managing resources and handling data to ensuring security and compliance. By leveraging cloud-native tools, implementing best practices, and addressing these challenges with targeted solutions, organizations can achieve efficient, scalable, and reliable ML workflows. This not only accelerates the development and deployment of ML models but also ensures their continuous improvement and adaptation to changing requirements.

5. Conclusion

Scaling machine learning (ML) model training with CI/CD pipelines in cloud environments offers significant advantages, including enhanced scalability, automation, and reliability. The integration of CI/CD practices with cloud-native tools and services facilitates efficient management of the entire ML lifecycle, from data ingestion and model training to deployment and monitoring. This survey has explored the key components of CI/CD pipelines tailored for ML, discussed the benefits and offerings of cloud environments, and addressed the challenges and solutions associated with scaling ML model training.

Implementing CI/CD pipelines for ML involves several best practices, such as modular pipeline design, automation of repetitive tasks, continuous testing, scalable infrastructure, robust monitoring, and adherence to security and compliance standards. Tools and frameworks like Kubeflow, MLflow, Jenkins, GitLab CI, and managed cloud services play a crucial role in streamlining these processes.

Despite the advantages, several challenges remain, including resource management, data handling, distributed training, model versioning, and security concerns. However, emerging solutions and innovative approaches are continuously being developed to overcome these hurdles, ensuring the effective and efficient scaling of ML workflows in cloud environments.

Future Research Directions

Future research in scaling ML model training with CI/CD pipelines in cloud environments can focus on several key areas. Advanced resource management techniques that leverage AI and ML to predict and allocate resources dynamically based on workload patterns and training requirements can optimize costs and improve efficiency. Integrating federated learning and edge computing with CI/CD pipelines could address data privacy concerns and reduce latency by bringing computation closer to the data source, enabling real-time ML applications. AI-driven automation of CI/CD pipelines could lead to more intelligent and adaptive workflows, reducing manual intervention and enhancing robustness. Standardizing CI/CD practices and tools across different cloud platforms could improve interoperability and simplify the integration of heterogeneous systems, making future research in developing universal standards and protocols vital. As data privacy regulations become more stringent, enhanced security and compliance frameworks specifically tailored for ML workflows in cloud environments will be crucial. This includes advanced encryption techniques, automated compliance checks, and robust access control mechanisms. Ensuring that ML models are explainable and fair is critical for ethical AI practices, and incorporating explainability and fairness checks into automated CI/CD pipelines can build trust and transparency in ML applications. Additionally, investigating sustainable AI practices that reduce the environmental impact of large-scale model training is essential. This involves exploring energy-efficient algorithms, optimizing resource utilization, and leveraging green cloud computing technologies. By addressing these research directions, the field can continue to advance, providing more effective and efficient methods for scaling ML model training with CI/CD pipelines in cloud environments. This will ultimately lead to more robust, reliable, and ethical AI systems that can adapt to the evolving needs of various industries.

References

1. Hummer, Waldemar, et al. "Modelops: Cloud-based lifecycle management for reliable and trusted ai." 2019 IEEE International Conference on Cloud Engineering (IC2E). IEEE, 2019.
2. Ma, Minghuang, et al. "Democratizing production-scale distributed deep learning." arXiv preprint arXiv:1811.00143 (2018).
3. Khandelwal, Mayank. A Service Oriented Architecture For Automated Machine Learning At Enterprise-Scale. MS thesis. 2018.
4. Iosup, Alexandru, et al. "The OpenDC vision: Towards collaborative datacenter simulation and exploration for everybody." 2017 16th International Symposium on Parallel and Distributed Computing (ISPDC). IEEE, 2017.
5. Felstaine, Eyal, and Ofer Hermoni. "Machine Learning, Containers, Cloud Natives, and Microservices." Artificial Intelligence for Autonomous Networks. Chapman and Hall/CRC, 2018. 145-164.
6. Arora, Akshay, et al. "Isthmus: secure, scalable, real-time and robust machine learning platform for healthcare." arXiv preprint arXiv:1909.13343 (2019).
7. Agrawal, Ashvin, et al. "Cloudy with high chance of DBMS: A 10-year prediction for Enterprise-Grade ML." arXiv preprint arXiv:1909.00084 (2019).
8. Demchenko, Yuri, et al. "Teaching DevOps and cloud based software engineering in university curricula." 2019 15th International Conference on eScience (eScience). IEEE, 2019.
9. Gamallo Gascón, Miguel. Design of a container-based microservices architecture for scalability and continuous integration in a solution crowdsourcing platform. Diss. Telecomunicacion, 2019.
10. Shah, Varun. "Towards Efficient Software Engineering in the Era of AI and ML: Best Practices and Challenges." INTERNATIONAL JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 3.3 (2019): 63-78.
11. Geigle, Chase, et al. "CLaDS: a cloud-based virtual lab for the delivery of scalable hands-on assignments for practical data science education." Proceedings of the 23rd Annual ACM Conference on Innovation and Technology in Computer Science Education. 2018.

12. Belmont, Jean-Marcel. Hands-On Continuous Integration and Delivery: Build and release quality software at scale with Jenkins, Travis CI, and CircleCI. Packt Publishing Ltd, 2018.
13. Arundel, John, and Justin Domingus. Cloud Native DevOps with Kubernetes: building, deploying, and scaling modern applications in the Cloud. O'Reilly Media, 2019.
14. Arundel, John, and Justin Domingus. Cloud Native DevOps with Kubernetes: building, deploying, and scaling modern applications in the Cloud. O'Reilly Media, 2019.
15. Arundel, John, and Justin Domingus. Cloud Native DevOps with Kubernetes: building, deploying, and scaling modern applications in the Cloud. O'Reilly Media, 2019.
16. Ward, David, and Chris Metz. "Role of Open Source, Standards, and Public Clouds in Autonomous Networks." Artificial Intelligence for Autonomous Networks. Chapman and Hall/CRC, 2018. 101-144.
17. Barabasi, Istvan. Framework and Patterns for Machine Learning as Microservices Using Open Source Tools and Open Data. Diss. Pace University, 2019.