International Journal of Core Engineering & Management

Volume-7, Issue-07, 2023, ISSN No: 2348-9510

MACHINE LEARNING MODEL SERVING AT SCALE: STRATEGIES, CHALLENGES, AND BEST PRACTICES IN CLOUD ENVIRONMENTS

Chandrakanth Lekkala
Chan.Lekkala@gmail.com

Abstract

With the ever-growing adoption of machine learning (ML), the necessity of scalable deployment and serving ML models at production levels has become more critical. Nevertheless, these issues introduce some stumbling blocks, mainly in Microsoft Azure's cloud platforms. This paper delves into the challenges of serving ML models at scale, encompassing deployment, versioning, surveillance, and improvement. It presents methods like containerization, A/B tests, and feature stores to tackle the indicated challenges. The paper also analyzes the best practices for deploying ML models on Azure and utilizes services such as Azure Machine Learning, Azure Kubernetes Service, and Azure DevOps. A case study is provided to illustrate the practical usability of such approaches in an actual situation. The findings emphasize the necessity of a holistic approach to model serving in ML, which involves DevOps practices and cloud-native technologies, among others, to ensure that these applications are scalable, reliable and continuously improved.

Keywords: Machine Learning, Model Serving, Scalability, Cloud Computing, Azure, Containerization, A/B Testing, Feature Store, DevOps

I. INTRODUCTION

Machine learning (ML) has become essential to modern software applications that empower businesses to make data-based decisions and derive insights from data [1] [47]. Nonetheless, deploying and inferring ML models at scale in a production environment creates specific challenges [2]. However, these challenges can be seen more clearly on cloud platforms, like Microsoft Azure, which has a vast set of services and tools for creating and deploying ML solutions [3]. Data mining techniques can play a significant role in analysing customer data for CRM [43].

This paper will investigate the problems that arise when modelling Machine Learning at a large scale. It will discuss techniques and best practices to resolve them, focusing mainly on the Azure cloud platform. The paper is structured as follows: Part 2 focuses on the main obstacles of ML model deployment with scale. Section 3 pinpoints the strategies that can be adopted to deal with these challenges, such as containerization, A/B testing, and feature stores. Part 4 covers how to deploy ML models on Azure. The demonstrative section 5 shows the practical application of these techniques through a case study. Last of all, Section 6 summarizes the paper and offers suggestions for further research.

II. CHALLENGES OF ML MODEL SERVING AT SCALE

Ensuring the reliability, scalability, and performance of ML models as they are scaled beyond the testing stage is a critical issue that must be tackled in the production environment where multiple challenges lie. Serving ML models at scale introduces several challenges around deployment, versioning, and performance [44]. Some of the key challenges are:

International Journal of Core Engineering & Management

Volume-7, Issue-07, 2023, ISSN No: 2348-9510

1. **Model deployment and versioning.**

Deploying ML models for production purposes is a big deal because there must be a robust process that is automated, consistent, and reproducible [4]. The model wraps up with its dependencies and the entire inference code, forming a deployable artefact, e.g., a container image [5]. Also, consider managing numerous model iterations and keeping the application with the correct model version implemented [6].

2. **Model surveillance and management.**

When deploying an ML model, it is essential to continuously monitor its performance and detect possible problems such as performance degradation or anomalies [7]. Different metrics statistics, such as latency and throughput, are essential, but error rates have to be prioritized and monitored [8]. Proactively addressing problems arguably helps to avoid conjuring up a negative image of the model and maintain its reliability and performance [9].

3. **Scalability and Resource Management**

Resource management is critical when the ML models are served at scale, significantly when the workloads vary, and the performance needs to be optimal at all times [10]. It includes a scaling approach that can be optionally used in high-concurrency situations [11]. By applying the resource-optimized allocation and auto-scaling features, the cost of service can be minimized without compromising performance levels [12].

4. **Model enhancement and retraining**

ML models are perpetually proficient because they must be paired and improved to adapt to changing data patterns and acquire accuracy on the fly [13]. This involves optimizing online retraining with new information, validating the model performance, and releasing the deployed batches of models to work with higher precision [14]. In addition, the lead-included model upgrades need tools like A/B testing to compare two models and choose the best one for production [15].

III. SERVING ML MODELS AT SCALE METHODS

Several techniques and strategies have been invented and integrated into industry practices to tackle the issue of serving ML models at scale [45]. Some of the essential methods are:

1. **Containerization**

Containerization, as a modern technology, has become a successful means of implementing and maintaining ML models structurally and reliably [16]. Containers are lightweight and isolated units of code that encapsulate model dependencies and inference code, among other things. Due to this, it is typical for machine learning models to migrate from the development to the testing server to the production environment without any functional issues [17].

Containerization, namely Docker and Kubernetes containers, has become the de facto standard for running and managing containerized applications, including ML models [18]. These recently emerging platforms offer the possibility of container orchestration, auto-scaling, and self-healing, which simplify the deployment and management of ML models on a rolling basis [19].

International Journal of Core Engineering & Management

Volume-7, Issue-07, 2023, ISSN No: 2348-9510

2. A/B Testing

A/B testing is a method that allows the running of two or more variations of the ML model that have already been deployed to observe the impact on performance [20]. Thus, it is possible to propose sending part of inbound requests to a new version of the model (vari ant B) to let the current version serve most of them (variant A) [21]. This enables the provision of the latest model functionality during the production environment operation [22].

Once we update ML models, we can use data to make further informed decisions through the A/B testing mentioned earlier [23]. By comparing key metrics and performance of the new and old model variants, such as accuracy, latency, and user engagement, the upgraded model can determine whether to be used or stick to what one is currently using [24]. If the new model works better than the existing system and can be implemented gradually, the risks caused by using unsatisfactory algorithm models in production will be eliminated.

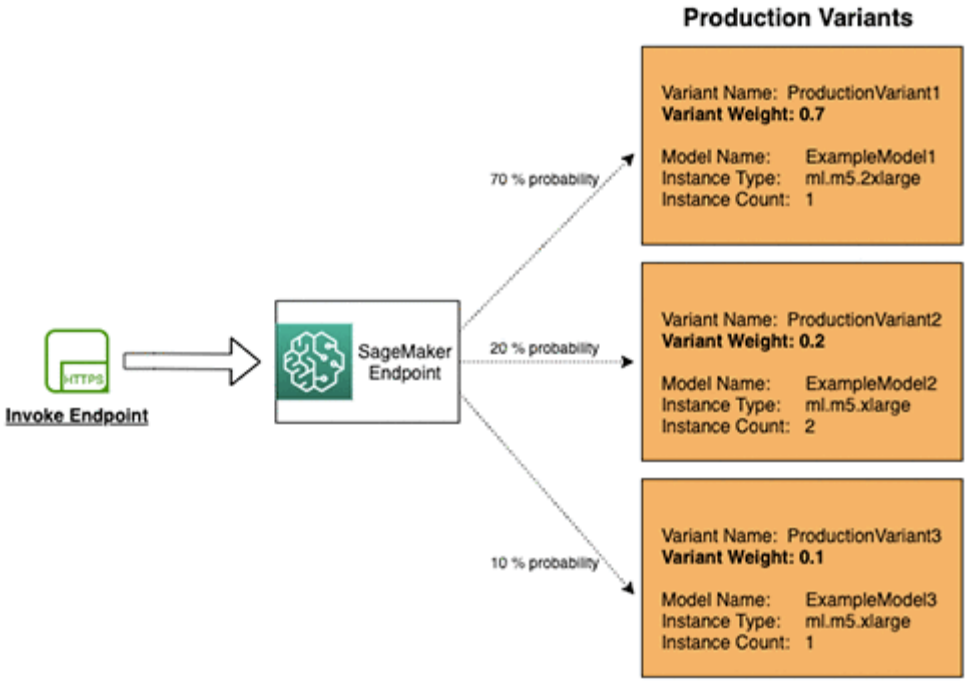

Figure 1: A/B testing workflow for comparing ML model variants

3. Feature Stores [48]

Feature stories are centralized repositories that store and manage the features used for training and serving ML models [25]. They provide a consistent and reusable way to access and serve features, enabling efficient model development and deployment [26]. Feature stores decouple the

International Journal of Core Engineering & Management

Volume-7, Issue-07, 2023, ISSN No: 2348-9510

feature engineering process from the model development process, allowing teams to collaborate and share features across different projects and models [27].

Using feature stores, ML teams can ensure that the same features are used for training and serving models, reducing the risk of data inconsistency and improving model performance [28]. Feature stores also enable feature versioning, allowing teams to track and manage feature changes over time [29]. This is particularly important when updating and enhancing ML models, as it ensures that they are trained and served with the most up-to-date and relevant features.

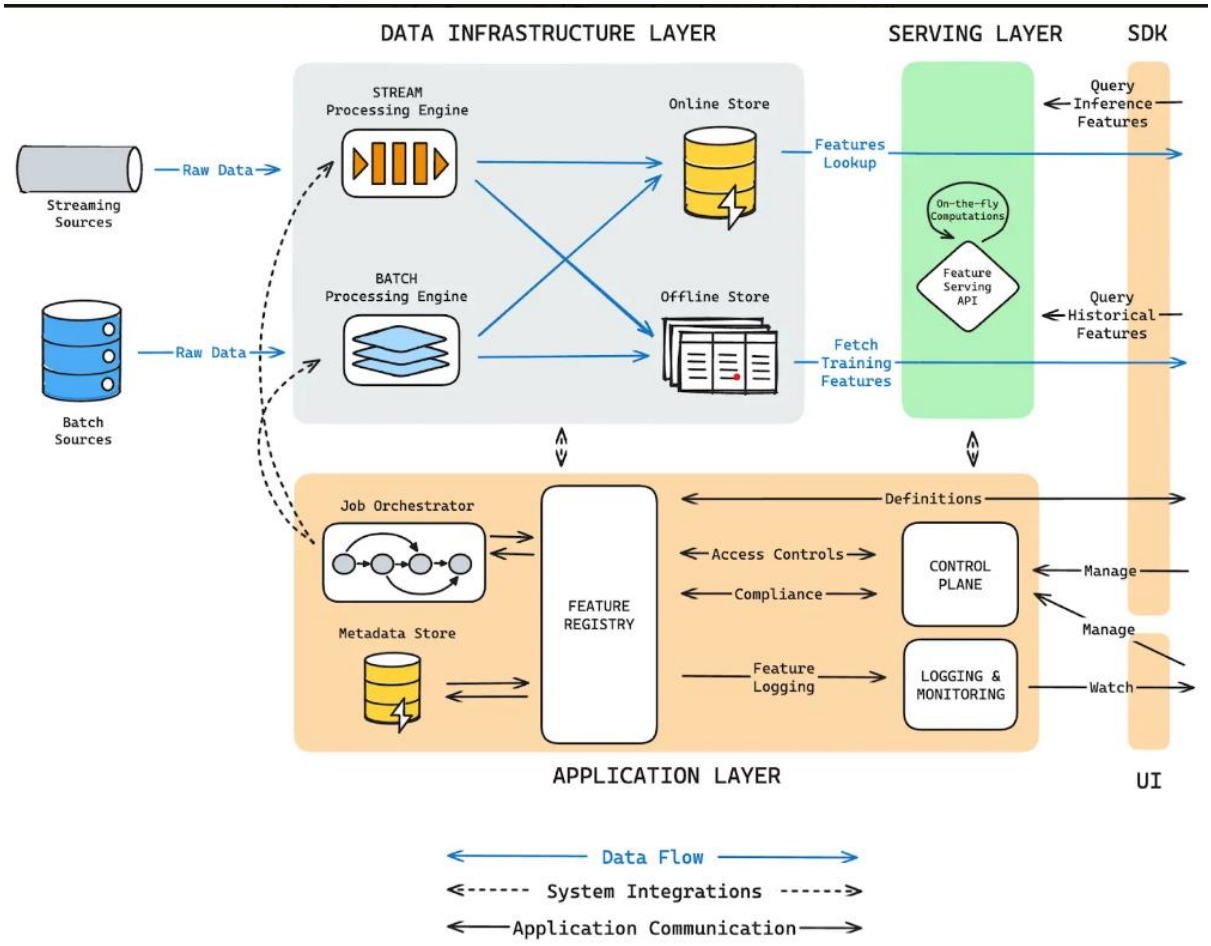

Figure 2: Feature store architecture for consistent feature management [49]

International Journal of Core Engineering & Management

Volume-7, Issue-07, 2023, ISSN No: 2348-9510

IV. BEST PRACTICES FOR DEPLOYING ML MODELS ON AZURE

Microsoft Azure offers a rich suite of AWS services built to construct, deploy, and manage ML solutions across the scale [29]. They provide a comprehensive review of different frameworks, techniques and tools used in big data environments [46]. Some of the best practices for deploying ML models on Azure are:

1. Explore Azure Machine Learning

Azure Machine Learning is a cloud-based platform that has the entire set of capabilities needed for developing, training, and deploying ML models, which is accomplished with the help of ML [30]. As an ML solution platform, it provides various options, from autoML to the versioning of models, and it gives deployment options to use with ease at scale [31]. Azure Machine Learning integrate with other Azure services, such as Azure Kubernetes Service and Azure Container Instances, conveniently while simultaneously offering the deployment and scaling of ML models [32].

2. Utilize Azure Kubernetes Services

Azure Kubernetes Service (AKS) gives an option for managed Kubernetes that does all the knotty work in seeding, running, and maintaining container applications, including the ML models [33]. AKS offers capabilities such as auto-scaling, self-repair, and integration with Azure Active Directory for Authentication and authorization [34]. The deployment of ML models to AKS is entirely viable as it offers an opportunity to capitalize on the advantages of Kubernetes, such as scalability and reliability, as well as integration with other Azure utilities [35].

3. Apply DevOps Principles

Using DevOps practices is necessary for creating and implementing machine learning models at scale since this can be achieved only using approaches relying on reliability and efficiency [36]. Azure DevOps is a composition of tools for implementing DevOps practices like workspace versioning, continuous integration and deployment (CI/CD), and monitoring at just a single click [37]. ML teams can reduce work overhead by automating model deployment, deploying reproducible models, and making continuous integration and delivery of model updates the norm [38].

Azure ML also interfaces with Azure DevOps, which permits the teams to craft CI/CD pipelines for model deployment and automate the telling of the model serving process [39]. Centering around this enmeshment, the data scientists can work seamlessly with the DevOps team, which assures a smooth transition from model development to production deployment.

V. CASE STUDY: ML MODEL DEPLOYMENT ON AZURE

The goal is to demonstrate the use of the techniques and best practices described in this paper. Let's consider deploying an ML model on Azure as an example.

Scenario: A retail company plans to implement a browsing and buying-related internet history-based recommendations system that will advise users on products. The ML model would learn using historical data from users, and its production deployment will be needed to immediately advise real-time customers.

International Journal of Core Engineering & Management

Volume-7, Issue-07, 2023, ISSN No: 2348-9510

Solution:

- The data science team creates and trains a recommendation model using Microsoft Azure Machine Learning. They use prebuilt ML techniques to optimize engineering and model versioning for track-to-track
- The built containerized using Docker is containerized, a cloud-based software container that provides portability and consistency across environments. The container image in Azure Container Registry has been chosen as a depot to make a secured distribution.
- The team develops its own CI/CD pipelines under Azure DevOps. This automated process includes creating, setting up, testing procedures, and distributing a containerized model version that facilitates a smooth deployment stream.
- The containerized environment runs on top of Azure Kubernetes Service (AKS), which is deployed on Azure for scalability and high availability. AKS approves a managed Kubernetes system, which then results in the efficient provision of orchestration and management of the model containers.
- An Azure ML-enabled A/B (split) testing is another stage in the deployment pipeline. This enables the task to have a clear picture of how the new model is compared to the existing one and empowers taking data-driven decisions.
- The feature store is integrated to handle and provide uniform features to uniform users for raining and serving purposes. It behaves as a decentralized repository. Its role includes rhythm and recurrences.
- Working with Azure Monitor, the model's performance metrics and detection of anomalies are monitored using the data received by the system. This provides insight into how the model has behaved by helping and predicting future issues with the model.
- Supply adequate information to users of the system so that they can take a proper decision.

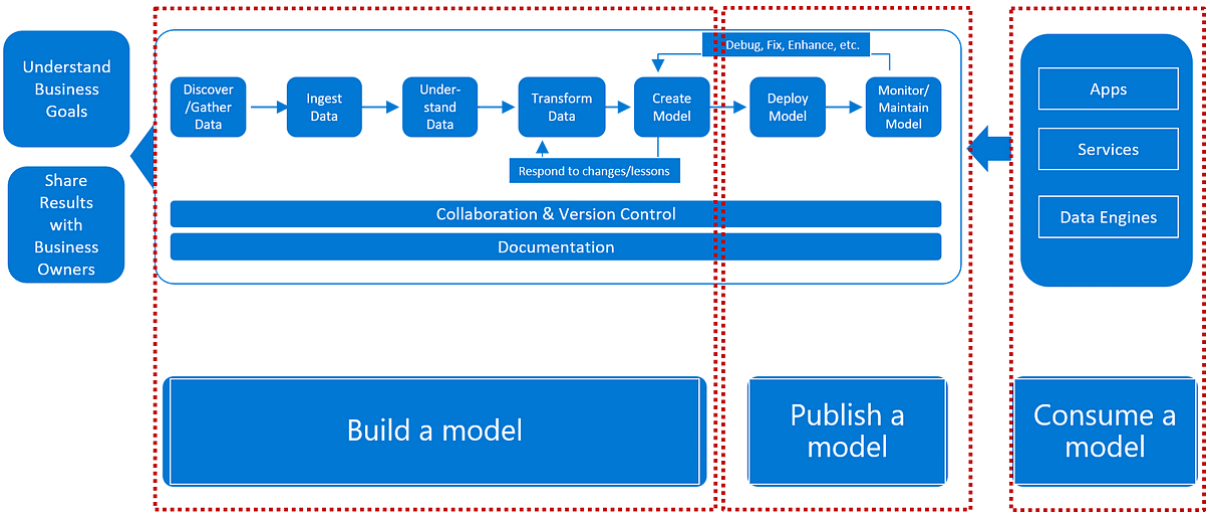

Figure 3: ML model deployment workflow on Azure

International Journal of Core Engineering & Management

Volume-7, Issue-07, 2023, ISSN No: 2348-9510

A retail company can achieve this through scaling and performance of the recommendation model, serving the function at scale with high performance, scalability, and reliability. Containerization, a/b testing, and feature store are used for model efficiency, especially during deployment, comparison, and improvement. Being tied up with Azure DevOps streamlines the deployment process and enables continuous delivery of model updates. The Product will be implemented using PHP. It should therefore run on any platform. We will use designs and code that are not specific to any platform, but we will primarily use Windows as the operating system.

VI. Resolution and Future Research.

Scaling up production environments with ML models necessitates addressing several issues – deployment, versioning, monitoring, and improvement – among others. The paper is based on related topics and suggested solutions like containerization, A/B testing, and feature stores. It also inspected the optimal methods of using ML models deployed to Azure via services such as Azure Machine Learning, Azure Kubernetes Service, and Azure DevOps.

Practical applications of the discussed techniques and best practices became a live example through the deployable Azure recommendation system. Integrating the modern DevOps approach and modern cloud technologies should be highly prioritized. They are one of the most effective ways of deploying and providing ML models at scale, performing in real-time and dynamically.

1. **Future research directions could include.**

a) **Advanced Model Monitoring and Anomaly Detection:**
Looking for innovative methods of real-time management of ML models arising in production is significant. Incorporating advanced anomaly detection algorithms, such as deep neural networks or unsupervised learning techniques, can facilitate active problem resolution and identification. In healthcare, patients have the right to access information about their illnesses and treatment options, which can feel like a burden for healthcare professionals [40].

b) **Serverless Computing for Model Deployment:**
The examination of the serverless computing world, Azure Functions, for smart deployment and service of ML models, may be developed as a research area. Serverless architectures make it easy and cost-effective to grow along with the workloads, which makes them more attractive for ML workloads. Investigating the perfect model packaging, cold start optimization, and the combination with other Azure services can give serverless computing the ML models implementation, which is worth [41].

c) **ML Model Serving on Edge Devices**
The ML model deployment within the edge computing architecture platform is to be investigated to enable real-time sensor information processing for both online and offline IoT scenarios. It is paramount to develop model compression, optimizing techniques, and security while embarking on edge devices where resources are limited. A study on fast communication and model synchronization between the models on the edge and cloud would help the apps run smoothly and improve the performance of ML applications in IoT environments [42].

The ML field continues pulsating as the latest technologies and advancements in implementing big-scale models are adopted. By tackling the challenges and using suitable instruments and platforms, organizations can apply ML or use it in production and orient businesses that acquire value and innovation.

REFERENCE

1. M.I. Jordan and T.M. Mitchell, "Machine learning: Trends, perspectives, and prospects," Science, vol. 349, no. 6245, pp. 255-260, Jul. 2015, doi: 10.1126/science.aaa8415.
2. D. Sculley et al., "Hidden technical debt in machine learning systems," Advances in neural information processing systems, vol. 28, June, 2015.
3. J. Langford, J. Pineau, A. Slivkins, and J.W. Vaughan, "The frontiers of machine learning," Communications of the ACM, vol. 62, no. 4, pp. 39-45, Apr. 2019, doi: 10.1145/3303869.
4. D. Crankshaw et al., "Clipper: A low-latency online prediction serving system," in 14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17), Mar. 2017, pp. 613-627.
5. M. Zaharia et al., "Accelerating the machine learning lifecycle with MLflow," IEEE Data Eng. Bull., vol. 41, no. 4, pp. 39-45, Dec. 2018.
6. K. Hazelwood et al., "Applied machine learning at Facebook: A data center infrastructure perspective," in 2018 IEEE International Symposium on High Performance Computer Architecture (HPCA), Feb. 2018
7. J. Klaise, A. Van Looveren, G. Vacanti, and A. Coca, "Monitoring and explainability of models in production," arXiv preprint arXiv:2007.06299, Jul. 2020.
8. D. Sato, A. Wider, and C. Windheuser, "Continuous delivery for machine learning," ThoughtWorks Inc., Aug. 2019.
9. E. Breck, S. Cai, E. Nielsen, M. Salib, and D. Sculley, "The ML test score: A rubric for ML production readiness and technical debt reduction," in 2017 IEEE International Conference on Big Data (Big Data), Dec. 2017
10. D. Huang, X. Liu, Q. Huang, and Z. Zheng, "Rapid Provisioning of Machine Learning Models via Efficient Knowledge Transfer," IEEE Access, vol. 7, pp. 28735-28745, Mar. 2019,
11. C. Olston et al., "Tensorflow-serving: Flexible, high-performance ml serving," arXiv preprint arXiv:1712.06139, Dec. 2017.
12. M. Boehm, A. Evfimievski, N. Pansare, and B. Reinwald, "Declarative machine learning-a classification of basic properties and types," arXiv preprint arXiv:1605.05826, May 2016.
13. T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 2016,
14. D. Peng et al., "Continuous Training for Neural Machine Translation," in Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), Nov. 2020,
15. E. Bakshy, D. Eckles, and M.S. Bernstein, "Designing and deploying online field experiments," in Proceedings of the 23rd International Conference on World Wide Web, Apr. 2014
16. D. Bernstein, "Containers and cloud: From lxc to docker to kubernetes," IEEE Cloud Computing, vol. 1, no. 3, pp. 81-84, Sep. 2014
17. C. Pahl, A. Brogi, J. Soldani, and P. Jamshidi, "Cloud container technologies: a state-of-the-art review," IEEE Transactions on Cloud Computing, vol. 7, no. 3, pp. 677-692, Jul. 2019

18. B. Burns, B. Grant, D. Oppenheimer, E. Brewer, and J. Wilkes, "Borg, omega, and kubernetes," Communications of the ACM, vol. 59, no. 5, pp. 50-57, Apr. 2016
19. K. Hightower, B. Burns, and J. Beda, "Kubernetes: up and running: dive into the future of infrastructure," O'Reilly Media, Sep. 2017.
20. X. Xu, H. Yu, and X. Pei, "A novel resource scheduling approach in container based clouds," in 2014 17th International Conference on Computational Science and Engineering, Dec. 2014
21. R. Kohavi, A. Deng, B. Frasca, T. Walker, Y. Xu, and N. Pohlmann, "Online controlled experiments at large scale," in Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Aug. 2013
22. D. Tang, A. Agarwal, D. O'Brien, and M. Meyer, "Overlapping experiment infrastructure: More, better, faster experimentation," in Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Jul. 2010
23. A. Fabijan, P. Dmitriev, H.H. Olsson, and J. Bosch, "Effective online controlled experiment analysis at large scale," in Proceedings of the 44th International Conference on Software Engineering, May 2018
24. E. Bakshy and E. Frachtenberg, "Design and analysis of benchmarking experiments for distributed internet services," in Proceedings of the 24th International Conference on World Wide Web, May 2015
25. D. Zhang, D.S. Berger, M. Harchol-Balter, and S. Shenker, "Bounding Delays in Stochastic Queuing Systems Using Stochastic Monotonicity," IEEE Transactions on Networking, vol. 28, no. 6, pp. 2724-2737, Dec. 2020
26. J. Hermann and M. Del Balso, "Meet michelangelo: Uber's machine learning platform," Uber Engineering Blog, Sep. 2017.
27. M. Palino, A. Jain, A. Bhardwaj, A. Tumanov, T. Zhu, and J.E. Gonzalez, "Runway: machine learning model experiment management tool," in Conference on Innovative Data Systems Research (CIDR), Jan. 2020.
28. H. Fang et al., "Ease. ml in 2021: Towards elastic, automatic, scalable and efficient machine learning systems," arXiv preprint arXiv:2111.10203, Nov. 2021.
29. D. Orr, G. Leszczynski, A. Arya, D. Lisuk, and S. Lee, "Managing machine learning projects with Amazon SageMaker Pipelines," Amazon Web Services Machine Learning Blog, Dec. 2020.
30. A. Agarwal et al., "Making contextual decisions with low technical debt," arXiv preprint arXiv:1606.03966, Jun. 2016.
31. E. Liberty et al., "Elastic machine learning algorithms in amazon sagemaker," in Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, Jun. 2020
32. O. Alipourfard, K. Sapra, Y. Yang, and M. Yu, "Towards Demystifying Serverless Machine Learning Training," in Proceedings of the 12th ACM Symposium on Cloud Computing, Nov. 2021
33. T. Chen and M. Li, "Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems," arXiv preprint arXiv:1512.01274, Apr. 2017.
34. Y. Peng, Y. Bao, Y. Chen, C. Wu, and C. Guo, "Optimus: an efficient dynamic resource scheduler for deep learning clusters," in Proceedings of the Thirteenth EuroSys Conference, Apr. 2018
35. W. Xiao et al., "Gandiva: Introspective cluster scheduling for deep learning," in 13th USENIX Symposium on Operating Systems Design and Implementation (OSDI 18), Oct. 2018

36. S. Alla and S.K. Adari, "DevOps: Continuous Delivery, Integration, and Deployment with DevOps," Packt Publishing, Mar. 2019.
37. J. Humble and D. Farley, "Continuous delivery: reliable software releases through build, test, and deployment automation," Pearson Education, Aug. 2010.
38. N. Forsgren, J. Humble, and G. Kim, "Accelerate: The science of lean software and DevOps," IT Revolution, Mar. 2018.
39. Microsoft Azure, "Machine learning DevOps with Azure ML," May 2023. [Online]. Available: https://docs.microsoft.com/en-us/azure/machine-learning/concept-model-management-and-deployment
40. D. Sculley et al., "Hidden technical debt in machine learning systems," Advances in Neural Information Processing Systems, vol. 28, pp. 2503-2511, 2015.
41. V. Ishakian, V. Muthusamy, and A. Slominski, "Serving deep learning models in a serverless platform," in 2018 IEEE International Conference on Cloud Engineering (IC2E), Apr. 2018,
42. S. Teerapittayanon, B. McDanel, and H.T. Kung, "Distributed deep neural networks over the cloud, the edge and end devices," in 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS), Jun. 2017
43. R. Chaudhary, S. Puthran, and N. Jain, "Data mining techniques for customer relationship management," International Journal of Core Engineering & Management, vol. 3, no. 6, pp. 44-51, Sep. 2016.
44. L. Baresi and G. Quattrocchi, "Training and serving machine learning models at scale," Lecture Notes in Computer Science, vol. 13611, pp. 669-683, Nov. 2022.
45. A. Gujarati, S. Elnikety, Y. He, K. S. McKinley, and B. B. Brandenburg, "Swayam: Distributed autoscaling to meet SLAs of machine learning inference services with resource efficiency," in Proceedings of the 18th ACM/IFIP/USENIX Middleware Conference, Dec. 2017, pp. 109-120.
46. A. Verma, A. H. Mansuri, and N. Jain, "A review on big data environment on different frameworks, techniques and tools," International Journal of Core Engineering & Management, vol. 3, no. 3, pp. 40-47, Jun. 2016.
47. D. Crankshaw and J. Gonzalez, "Prediction-serving systems," Queue, vol. 16, no. 1, pp. 70-93, Jan.-Feb. 2018.
48. Amazon Web Services. (2020). *A/B Testing ML models in production using Amazon SageMaker | Amazon Web Services*
49. Qwak.com. (2024). *Feature Store Architecture and How to Build One | Qwak*.
50. Godfried, I. (2018). *Deploying deep learning models: Part 1 an overview - Towards Data Science*. [online] Medium.