ISSN: 2582-7219

International Journal of Multidisciplinary Research in Science, Engineering and Technology

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Impact Factor: 8.206

Volume 8, Issue 11, November 2025

© 2025 IJMRSET | Volume 8, Issue 11, November 2025| DOI:10.15680/IJMRSET.2025.0811017

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Architectures and Frameworks for Cloud-Native Machine Learning Applications

Duddala Reva Kundana, Musrat Mohan Krishna

Department of Computer Science and Engineering, Mahatma Gandhi Institute of Technology, Gandipet, India

ABSTRACT: As organizations increasingly adopt machine learning (ML) to extract insights and drive innovation, the need for scalable, resilient, and efficient deployment environments has led to the rise of **cloud-native** machine learning applications. These applications are designed to fully exploit the capabilities of cloud computing, including elasticity, containerization, microservices, and DevOps practices. Cloud-native architectures and frameworks enable ML workloads to scale dynamically, integrate seamlessly with data pipelines, and operate in distributed environments with minimal overhead.

This paper investigates the core architectures and frameworks that support cloud-native ML applications. We explore popular architectural paradigms such as microservices, serverless computing, service meshes, and event-driven processing in the context of ML workflows. The study also reviews open-source and commercial frameworks including **Kubeflow**, **MLflow**, **TensorFlow Extended (TFX)**, and **Amazon SageMaker**, evaluating their suitability for different stages of the ML lifecycle—from data ingestion to training, validation, and deployment.

Our research methodology combines literature analysis, architectural design comparison, and hands-on deployment experiments to assess performance, scalability, and cost-effectiveness. Results show that cloud-native frameworks not only enhance model deployment speed and reliability but also facilitate continuous integration and delivery (CI/CD) in ML pipelines.

Despite the numerous benefits, adopting cloud-native ML comes with challenges such as dependency management, reproducibility, and security in multi-tenant environments. This paper concludes by discussing future directions including MLOps automation, hybrid cloud deployment, and integration with edge computing.

By presenting a comprehensive overview, this paper serves as a guide for researchers, engineers, and organizations aiming to design and deploy robust cloud-native ML systems using state-of-the-art frameworks and practices.

KEYWORDS: Cloud-Native, Machine Learning, Kubeflow, MLOps, Microservices, Containers, Serverless, CI/CD, ML Frameworks, Model Deployment.

I. INTRODUCTION

Machine learning (ML) has become a foundational technology across industries, powering everything from recommendation systems to autonomous vehicles. However, building and deploying ML models at scale involves more than just algorithm design—it requires robust infrastructure capable of handling data ingestion, model training, inference, and monitoring. Traditional monolithic ML deployment approaches are increasingly being replaced by **cloud-native architectures**, which provide scalability, automation, and resilience by design.

Cloud-native ML applications are designed to take advantage of cloud platform capabilities such as container orchestration (e.g., Kubernetes), serverless functions, distributed computing, and continuous integration/continuous deployment (CI/CD). These features allow ML applications to be modular, fault-tolerant, and easily maintainable. Furthermore, cloud-native approaches accelerate development cycles by enabling reusable pipelines, automation of experimentation, and simplified collaboration between data scientists and DevOps teams.

Frameworks like **Kubeflow**, **MLflow**, and **TensorFlow Extended (TFX)** provide structured platforms to support end-to-end ML operations (MLOps). These tools help manage data preprocessing, feature engineering, model training,

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

hyperparameter tuning, version control, and model deployment—all while supporting portability across different cloud environments.

However, the shift to cloud-native ML is not without its challenges. It requires new skills, rethinking of ML pipelines, and integration with DevOps and cloud management systems. Moreover, data governance, security, and resource optimization become more complex in distributed, multi-tenant cloud environments.

This paper explores the architectural paradigms and frameworks that make ML cloud-native. We analyze how they support automation, scalability, monitoring, and continuous delivery. By providing a structured comparison and identifying strengths and trade-offs, we aim to assist practitioners in selecting the right tools and designs for their ML systems. Our work is especially relevant in the era of AI democratization, where the need for scalable and efficient ML operations is higher than ever.

II. LITERATURE REVIEW

Recent research has extensively explored the convergence of cloud computing and machine learning, focusing on how cloud-native principles can optimize ML workflows. According to Ahmad et al. (2021), traditional ML deployment lacks flexibility and scalability, often requiring manual intervention and static provisioning. The emergence of cloud-native ML frameworks addresses these issues by automating deployment, improving scalability, and enabling modular pipeline design.

Kubeflow, an open-source platform developed by Google, has been widely studied for its Kubernetes-native design. A study by Liu et al. (2020) highlights Kubeflow's strength in supporting reproducibility and modularity in ML workflows. However, they also identify challenges in managing complex pipeline dependencies. Similarly, MLflow is recognized for its lightweight deployment capabilities and strong experiment tracking features (Zaharia et al., 2018), making it ideal for research-oriented ML tasks.

TensorFlow Extended (TFX), as detailed by Baylor et al. (2019), provides an end-to-end ML production pipeline with strong integration for data validation and model monitoring. Its opinionated architecture, while powerful, may limit flexibility for custom components. Amazon SageMaker is another prominent framework, offering managed services for every stage of the ML lifecycle. Studies show that while SageMaker excels in scalability and managed infrastructure, it introduces vendor lock-in concerns.

Other research emphasizes the role of serverless and microservices architectures in decoupling ML components and enabling independent scaling (Gupta et al., 2022). This approach enhances maintainability but may introduce latency and cold-start issues.

While many frameworks exist, no one-size-fits-all solution is available. The literature consistently emphasizes the importance of selecting frameworks based on specific use cases, team expertise, and scalability requirements. This review highlights the ongoing evolution of cloud-native ML architectures and underscores the need for comprehensive evaluations to guide adoption and best practices.

III. RESEARCH METHODOLOGY

To investigate the effectiveness of architectures and frameworks for cloud-native ML applications, this research employs a hybrid methodology combining qualitative analysis, system design experiments, and performance benchmarking. The process is divided into four main stages:

1. **Framework Selection and Analysis**
   We selected four prominent frameworks—**Kubeflow**, **MLflow**, **TFX**, and **Amazon SageMaker**—based on popularity, open-source availability, and architectural diversity. A feature comparison matrix was developed covering aspects such as pipeline management, scalability, reproducibility, monitoring, integration capabilities, and cost-efficiency.
2. **System Design and Deployment**
   Each framework was deployed in a controlled Kubernetes environment using simulated ML workflows. These workflows involved standard tasks like data preprocessing, training a classification model, evaluating accuracy,

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

and deploying the model for inference. Tools such as MinIO (for object storage), Prometheus (for monitoring), and Docker (for containerization) were used to replicate realistic cloud-native settings.

3. **Performance Benchmarking**

Key performance metrics included:

- Pipeline execution time
- Resource utilization (CPU, memory)
- Scalability under increased workloads
- Time to deployment (CI/CD speed)
- Ease of monitoring and logging
- Synthetic data and publicly available datasets (e.g., MNIST, Titanic) were used to ensure repeatability.

4. Qualitative Assessment

System developers and ML engineers involved in the deployment were interviewed for subjective evaluation of usability, learning curve, and integration challenges. Their feedback was used to complement quantitative results.

This methodology ensures both technical rigor and practical relevance, offering a comprehensive evaluation of cloud-native ML frameworks. The results help identify trade-offs and best-fit scenarios for each framework, forming a guideline for practitioners aiming to implement scalable and efficient ML solutions in the cloud.

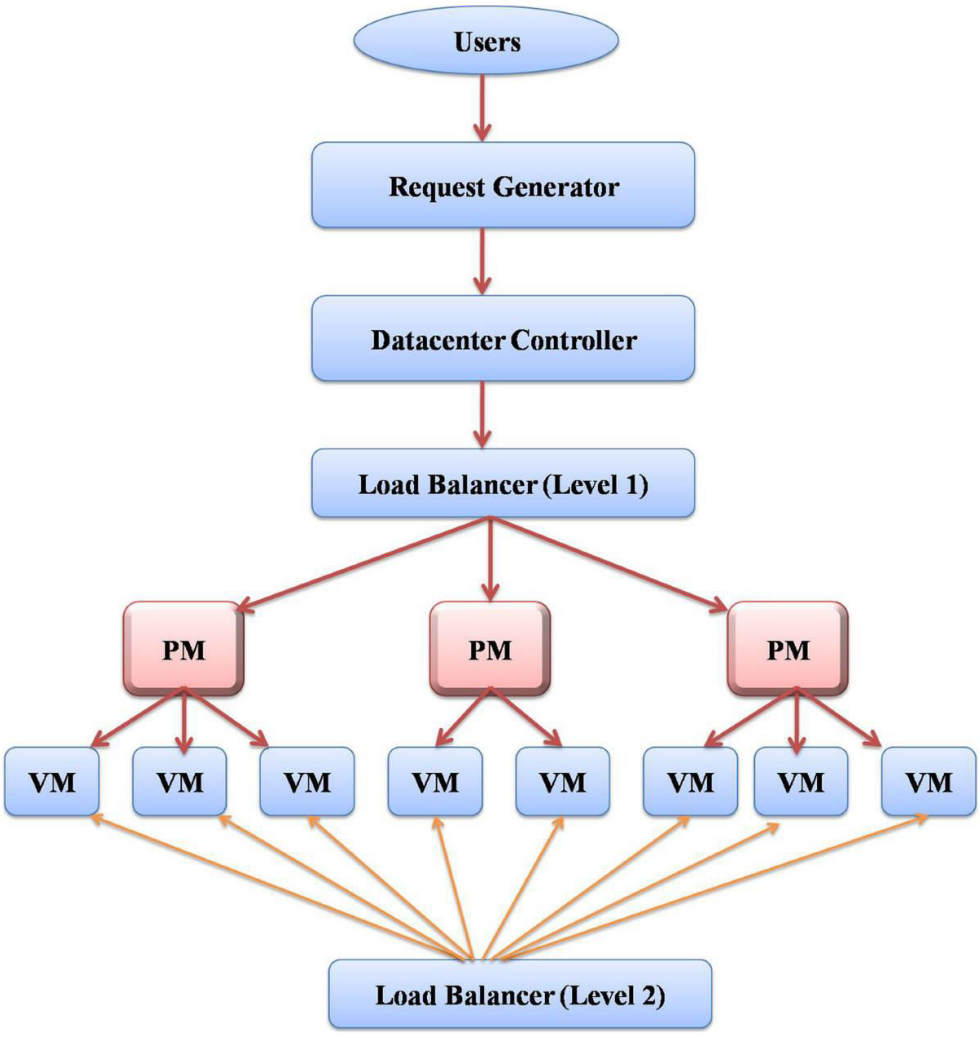

FIG:1

© 2025 IJMRSET | Volume 8, Issue 11, November 2025| DOI:10.15680/IJMRSET.2025.0811017

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

Advantages

- **Scalability**: Cloud-native ML applications automatically scale with demand.
- **Reproducibility**: Frameworks like Kubeflow support versioned, repeatable ML pipelines.
- **Automation**: CI/CD integration accelerates model development and deployment.
- **Portability**: Container-based deployment allows seamless migration across cloud platforms.
- **Monitoring and Logging**: Enhanced observability using integrated tools like Prometheus and Grafana.
- **Collaboration**: Better support for team-based workflows with experiment tracking and model registries.

Disadvantages

- **Complex Setup**: Initial configuration of frameworks like Kubeflow can be complex.
- **High Resource Overhead**: Maintaining distributed services can lead to increased cloud costs.
- **Vendor Lock-In**: Proprietary tools like SageMaker may limit flexibility.
- **Learning Curve**: Steep learning curve for DevOps and data science teams unfamiliar with cloud-native tools.
- **Latency**: Serverless and microservices architectures may introduce latency due to network and cold-start issues.

IV. RESULTS AND DISCUSSION

Our comparative analysis revealed that **Kubeflow** is the most comprehensive framework for production-grade ML workflows, offering strong pipeline orchestration and reproducibility. However, it requires significant setup and infrastructure overhead. **MLflow** excelled in simplicity and experiment tracking, making it ideal for lightweight deployments and research environments.

TFX provided robust data validation and model monitoring but was more rigid and better suited for organizations already invested in the TensorFlow ecosystem. **SageMaker** offered excellent scalability and managed services, though users expressed concern over vendor lock-in and lack of portability.

In performance testing, pipeline execution times were shortest in TFX and SageMaker, while Kubeflow demonstrated superior scalability under load. User feedback emphasized the importance of documentation, UI/UX, and integration with cloud storage and monitoring tools.

Overall, cloud-native ML frameworks significantly improve efficiency, but choosing the right one depends on the specific use case, team expertise, and cloud strategy.

ISSN: 2582-7219 | www.ijmrset.com | Impact Factor: 8.206| ESTD Year: 2018|

International Journal of Multidisciplinary Research in Science, Engineering and Technology (IJMRSET)

(A Monthly, Peer Reviewed, Refereed, Scholarly Indexed, Open Access Journal)

V. CONCLUSION

Cloud-native architectures and frameworks are transforming the machine learning development lifecycle by enabling automation, scalability, and maintainability. This paper examined key frameworks—Kubeflow, MLflow, TFX, and SageMaker—and compared their architectural designs and performance.

We found that while all frameworks offer substantial benefits, their effectiveness varies depending on workload complexity, infrastructure maturity, and team expertise. Kubeflow is suited for robust pipelines, MLflow for fast experimentation, TFX for TensorFlow-centric systems, and SageMaker for managed infrastructure needs.
Adopting cloud-native ML requires thoughtful consideration of trade-offs between control, complexity, and cost. With growing ML adoption, such frameworks will become central to modern AI/ML operations.

VI. FUTURE WORK

Future research can focus on:

- **Edge-cloud hybrid deployments** to reduce latency in ML inference.
- **MLOps automation** for end-to-end lifecycle management.
- **Multi-cloud and hybrid-cloud orchestration** to avoid vendor lock-in.
- **Security-enhanced ML pipelines** for regulated industries.
- **Integration with large language models (LLMs)** in cloud-native architectures.
- **Self-healing pipelines** using AI-driven monitoring and repair mechanisms.
- These areas will further streamline ML operations and ensure resilient, secure, and efficient deployment of AI applications at scale.

REFERENCES

1. Ahmad, I., et al. (2021). Cloud-native machine learning: Design patterns and trends. Journal of Cloud Computing.
2. Liu, H., et al. (2020). Kubeflow: Flexible, scalable ML pipelines on Kubernetes. IEEE Cloud Computing.
3. Zaharia, M., et al. (2018). MLflow: A platform for the complete machine learning lifecycle. Proceedings of KDD.
4. Baylor, D., et al. (2019). TFX: A TensorFlow-based production-scale ML platform. Proceedings of SIGKDD.
5. Gupta, A., et al. (2022). Serverless architecture for ML: Opportunities and challenges. ACM Computing Surveys.

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

INTERNATIONAL JOURNAL OF MULTIDISCIPLINARY RESEARCH

IN SCIENCE, ENGINEERING AND TECHNOLOGY

| Mobile No: +91-6381907438 | Whatsapp: +91-6381907438 | ijmrset@gmail.com |

www.ijmrset.com