International Journal of Scientific Research in Science, Engineering and Technology

Print ISSN - 2395-1990
Online ISSN : 2394-4099

OPEN ACCESS

Available Online at :www.ijsrset.com
doi : https://doi.org/10.32628/IJSRSET25121179

Developing an AI Model Registry and Lifecycle Management System for Cross-Functional Tech Teams

Toluwase Peter Gbenle[1], Abraham Ayodeji Abayomi[2], Abel Chukwuemeke Uzoka[3], Oyejide Timothy Odofin[4], Oluwasanmi Segun Adanigbo[5], Jeffrey Chidera Ogeawuchi[6]

[1]NICE Ltd Nexidia, Atlanta, Georgia, USA

[2]SKA Observatory, Macclesfield, UK

[3]The Vanguard group Charlotte, North Carolina, USA

[4]DXC Technology, Poland

[5]Independent Researcher, Delaware, USA

[6]Megacode Company, Dallas Texas, USA

ARTICLEINFO

Article History:

Accepted : 11 Aug 2024

Published: 20 Aug 2024

Publication Issue :

Volume 11, Issue 4

July-August-2024

Page Number :

442-456

ABSTRACT

This paper presents a comprehensive solution for managing AI models across their lifecycle through the development of an AI model registry and lifecycle management system. As AI continues to play a crucial role across industries, the complexity of managing models—from development to deployment—presents significant challenges, especially within cross-functional teams. These challenges include issues such as model versioning, metadata management, deployment inconsistencies, and communication breakdowns among data scientists, engineers, and business stakeholders. The proposed system addresses these challenges by providing a centralized platform that integrates features such as version control, metadata management, and automated deployment, thereby improving transparency and reducing the risk of deployment errors. Furthermore, the system fosters enhanced collaboration by integrating widely-used project management tools like GitHub, Jira, and Slack, ensuring that teams remain aligned throughout the model's lifecycle. By enabling continuous monitoring and incorporating automated model drift detection, the system ensures that AI models remain accurate and efficient post-deployment. This paper also explores the technical implementation strategy for the system, including the use of containerization, cloud-native infrastructure, and microservices architecture to ensure scalability and flexibility. The implications of this work extend beyond technical considerations, as it enhances collaboration, improves model quality, and accelerates deployment cycles. Future research directions include exploring automation in model updates, scalability in large enterprises, and the integration of additional tools and frameworks. This work provides a critical step toward optimizing AI model management, offering a scalable, efficient, and secure approach to managing AI models throughout their lifecycle.

Keywords: AI Model Management, Lifecycle Management, Cross-Functional Teams, Version Control, Model Deployment, Collaboration Tools

INTRODUCTION

1.1 Context and Relevance

Artificial Intelligence (AI) is rapidly transforming industries across the globe, ranging from healthcare and finance to retail and manufacturing. The increasing reliance on AI models for decision-making, automation, and customer personalization has made the management of these models an essential part of organizational workflows [1, 2]. Cross-functional teams—comprising data scientists, engineers, and business leaders—often face significant challenges in coordinating their efforts to develop, deploy, and maintain these AI models [3, 4]. The complexity of managing different versions of AI models, coupled with the evolving nature of machine learning techniques, makes this task increasingly difficult. As organizations scale their AI initiatives, it becomes imperative to establish effective management strategies that streamline these processes while ensuring alignment across diverse teams [5, 6].

Moreover, as AI models become more integral to decision-making processes, the need for reliable management systems becomes clearer [7]. The process of managing AI models is not only about ensuring their accuracy but also about keeping track of model versions, monitoring their performance in production environments, and ensuring compliance with regulatory standards [8, 9]. This calls for sophisticated systems that can handle the full lifecycle of an AI model, from development and validation to deployment and retirement. In this context, understanding how cross-functional teams can collaborate effectively while managing these models is critical [10-12].

In industries such as autonomous vehicles or medical diagnostics, mismanagement of AI models can have disastrous consequences [13]. As such, an AI model registry and lifecycle management system that facilitates seamless collaboration and version control is essential for minimizing errors, improving model performance, and maintaining regulatory compliance [14, 15]. This paper aims to address these challenges and propose a robust solution to support the management of AI models, particularly in cross-functional environments.

1.2 Problem Statement

AI model management faces a variety of challenges that often hinder the smooth operation of cross-functional teams [16]. One of the most significant challenges is version control, where different teams may work on divergent versions of the same model. This lack of uniformity can lead to inconsistent outcomes, confusion, and inefficiencies in both model development and deployment [17]. As AI models evolve over time, maintaining a record of changes and updates becomes increasingly complex, especially when multiple teams are involved. Without an effective versioning system, it becomes difficult to track model performance, replicate results, or rollback to previous versions when issues arise [18].

Another critical issue in AI model management is deployment. While models may perform well in development environments, translating that success to production can be difficult [16, 19]. Models need to be integrated with existing systems, and their deployment often requires careful orchestration between various tech teams [20]. These teams must ensure that the model performs optimally under real-world conditions, which may involve managing multiple versions of models across different environments [21]. Often, the lack of a centralized registry complicates the process, leading to deployment delays or errors, particularly when new models must be integrated into legacy systems or when scaling models across multiple cloud platforms [21].

Collaboration is also a significant challenge, particularly in large organizations where teams are spread across different geographies. When data scientists, engineers, and business analysts are not effectively communicating, it can lead to misalignment in the goals and expectations surrounding AI models [22]. Without a unified framework for model management, each team may operate in silos, which ultimately hinders the overall efficiency of AI projects. Therefore, a centralized model registry that supports seamless communication and collaboration between different teams is crucial in overcoming these challenges [23].

1.3 Objectives and Scope

This paper aims to develop a conceptual framework for an AI model registry and lifecycle management system tailored for cross-functional teams. The goal is to create a system that facilitates smooth collaboration, enables version control, and supports the deployment and monitoring of AI models throughout their lifecycle. By addressing the challenges identified in model management, the system should aim to enhance transparency, reduce errors, and improve efficiency in AI development and deployment processes. The scope of the paper focuses on providing a comprehensive system that can be adapted to various industries, ensuring its relevance across multiple sectors.

The proposed AI model registry will serve as a centralized repository for managing the various versions of AI models, tracking their updates, and ensuring that the right models are deployed in the appropriate environments. Additionally, the lifecycle management system will incorporate tools for monitoring models in production, collecting performance data, and enabling continuous improvement. By integrating version control and deployment management, the system will address key pain points in the AI model management process. Furthermore, the registry will provide cross-functional teams with a collaborative platform that streamlines communication and decision-making.

This paper will explore the design, implementation, and potential impact of such a system on cross-functional collaboration and model management. Through a review of current practices, tools, and technologies, it will present a robust framework that organizations can use to optimize their AI model management processes. The ultimate objective is to contribute to the efficient and effective development of AI systems that can be deployed and maintained at scale while promoting inter-team collaboration and reducing operational risks.

Background and Literature Review

2.1 AI Model Management

AI model management is a critical component of ensuring that machine learning models perform optimally and remain reliable throughout their lifecycle [16, 24]. Recent literature highlights the increasing complexity of managing AI models as they evolve from development to deployment, requiring robust systems to track versions, manage metadata, and ensure model compliance [25]. AI model registries, such as MLflow and ModelDB, have emerged as popular tools for addressing these challenges [24, 26]. These platforms offer centralized storage for models, allowing teams to easily track changes, test versions, and retrieve prior models when necessary [26]. However, despite their utility, many registries still lack advanced integration with deployment systems, which can create silos between development and production environments [16, 24].

Monitoring AI models is another critical aspect of model management. Once a model is deployed in production, it must be monitored continuously to detect performance degradation, model drift, or any unexpected behavior [16, 27]. Studies emphasize the importance of real-time monitoring systems that can trigger alerts when models begin to underperform, allowing for timely intervention [28, 29]. Tools like Prometheus and Grafana are commonly used for this purpose, but they often require custom integration with machine learning pipelines, which can be resource-intensive [25]. Furthermore, monitoring in production environments remains underdeveloped in terms of automated feedback loops that can help improve model performance over time [30, 31].

Deployment practices in AI model management also come with challenges. Deploying models often requires coordination between various teams, including data scientists who develop the models, engineers who integrate them into systems, and business leaders who need to ensure the models meet organizational goals [32]. Current deployment tools, such as Kubernetes and Docker, are widely used to handle scaling and infrastructure concerns. However, literature points out that there is a lack of standardized best practices for deploying AI models at scale, particularly in hybrid cloud environments [16, 33]. Moreover, ensuring that models are consistently deployed across different environments without introducing errors remains a significant challenge in large-scale AI implementations [32, 34].

2.2 Challenges in Cross-Functional Collaboration

Cross-functional collaboration between diverse teams is fundamental for the successful deployment and operation of AI models. Data scientists, engineers, and business stakeholders each bring unique perspectives and expertise, but often face communication barriers that can hinder the overall efficiency of AI projects [35, 36]. Data scientists typically focus on developing accurate models, while engineers are tasked with ensuring that these models can be efficiently deployed and scaled within the organization's infrastructure [17, 37]. Business stakeholders, on the other hand, are concerned with the real-world impact of AI models, including their alignment with business goals and customer needs. Without effective communication and shared objectives, these diverse priorities can lead to misalignment and inefficiencies [38, 39].

A key challenge in this collaboration is the lack of a unified framework for model management. Without a centralized system to track the lifecycle of AI models, teams often work in silos, leading to discrepancies in version control, deployment timelines, and performance monitoring [19, 21]. Literature indicates that these silos are exacerbated by the rapid pace of AI development, where changes to models or infrastructure may not be communicated promptly to other teams [40]. Additionally, data scientists may not fully understand the engineering requirements for scaling models, while engineers may lack the domain-specific knowledge needed to optimize models effectively [16]. A robust AI model registry and lifecycle management system can bridge these gaps by providing a single platform for collaboration, ensuring that all teams have access to the same information and are aligned on model updates and performance metrics [37].

Moreover, the cultural and organizational challenges of cross-functional collaboration can also affect AI model management. In many organizations, the division between technical and non-technical teams can lead to a lack of understanding regarding the importance of specific model management practices [41-43]. For instance, business stakeholders may not appreciate the significance of proper version control or the need for continuous model monitoring, leading to suboptimal resource allocation [44, 45]. Creating an inclusive environment where all teams are encouraged to collaborate on AI model management can help overcome these challenges. Literature suggests that when business and technical teams work together from the outset, AI models are more likely to meet both operational and strategic goals, enhancing their long-term success [41, 42].

2.3 Lifecycle Management Systems

AI lifecycle management systems are designed to handle the full range of activities involved in developing, deploying, and maintaining AI models. Current systems typically offer tools for version control, model training, testing, and monitoring, but many still face limitations in terms of integration, scalability, and user-friendliness [20]. Tools like TensorFlow Extended (TFX) and Kubeflow are examples of open-source platforms that provide end-to-end lifecycle management capabilities, enabling teams to automate processes from data ingestion to model deployment [46-48]. However, these systems often require substantial expertise to set up and maintain, and they may not be suitable for smaller organizations or teams with limited technical resources [49].

One of the most significant limitations of existing lifecycle management tools is their inability to integrate with other platforms used by cross-functional teams seamlessly [50]. For example, while TFX and Kubeflow are powerful for automating machine learning pipelines, they are often disconnected from other systems like project management tools or business intelligence platforms, which could facilitate better collaboration [50, 51]. This lack of integration between different tools and systems can lead to inefficiencies, with teams manually transferring data between systems or failing to keep all stakeholders up to date on model changes [52-54]. Literature suggests that overcoming this issue requires developing integrated platforms that support the full AI lifecycle, from model development to deployment and post-deployment monitoring, within a single, user-friendly interface [55].

Another limitation of current lifecycle management systems is their focus on technical aspects, often neglecting the business and operational side of AI deployment. While many tools offer excellent technical features, they often fail to provide sufficient insight into how models align with business goals or how their performance impacts key metrics [20, 56]. AI lifecycle management must evolve to include business-driven features that allow stakeholders to track the ROI of models, assess their impact on business outcomes, and ensure compliance with regulatory standards [57, 58]. By doing so, organizations can make more informed decisions about the deployment and scaling of AI models. In conclusion, while current lifecycle management tools offer valuable capabilities, there is a significant opportunity to develop more integrated, business-aware systems that can meet the evolving needs of cross-functional teams [25].

Proposed System Design

3.1 AI Model Registry

The proposed AI model registry is designed as a centralized, cloud-based platform that serves as the single source of truth for all AI models throughout their lifecycle. At its core, the registry supports advanced version control, enabling teams to track changes across different iterations of a model. Each version of a model is cataloged with associated metadata, including details about model architecture, training data, performance metrics, and deployment environment [59, 60]. This metadata management ensures that teams can easily identify and retrieve specific model versions based on their requirements, whether for debugging, testing, or production deployment. The system allows teams to maintain a clear history of updates, ensuring transparency and reproducibility of results across various iterations [61, 62].

A critical feature of the registry is its access control functionality. By implementing role-based access control (RBAC), the system restricts access to specific model versions, metadata, and other sensitive information based on the user's role within the organization [60, 62]. This ensures that only authorized personnel can make critical changes or deploy models, minimizing the risk of unauthorized modifications or accidental deployment of incorrect versions [16, 21]. Access control is also designed to foster secure collaboration between teams, allowing data scientists to freely experiment with model versions while ensuring that engineers and business stakeholders have access to the necessary information for their roles [52, 53]. In addition, audit logs are maintained to track all actions performed on models within the registry, providing an additional layer of security and accountability [61, 63].

The architecture of the registry is designed to be scalable and easily integrated with existing machine learning platforms and infrastructure. By leveraging cloud-based storage solutions like Amazon S3 or Google Cloud Storage, the registry can handle large volumes of data and support high availability [64, 65]. Additionally, the registry's API layer allows for seamless integration with tools used by cross-functional teams, such as Jenkins for continuous integration/continuous deployment (CI/CD), Jira for project management, and Slack for communication. This level of integration ensures that the registry is not a standalone system but rather an integral part of the broader AI development ecosystem [66].

3.2 Lifecycle Management Workflow

The AI model lifecycle management workflow begins with model development, where data scientists and engineers collaborate to define the model's objectives, data sources, and training processes. This stage includes data preprocessing, feature engineering, model selection, and training [16, 20]. The model registry plays a key role in this stage by allowing teams to store different versions of the model as they experiment with various algorithms and hyperparameters. Each iteration is tracked with metadata to document the decisions made and the model's performance during training, ensuring that progress is recorded and that models can be easily compared.

Once the model is trained and evaluated, it moves to the deployment phase. The registry enables the selection of the best-performing model version, which is then deployed into a staging or production environment. Automated deployment tools, such as Kubernetes or Docker, can be integrated with the registry to streamline this process, ensuring that the correct version of the model is deployed without manual intervention [67, 68]. The lifecycle management system also allows for automated rollback to previous versions in case of issues, minimizing downtime and disruptions. Continuous integration and delivery (CI/CD) pipelines facilitate seamless updates and deployment, ensuring that new model versions can be pushed to production as soon as they pass testing stages [69, 70].

The final phase of the workflow involves monitoring and maintaining the deployed models. Once in production, the model's performance is continuously tracked, and metrics such as accuracy, latency, and system resource usage are collected. This monitoring can be facilitated through integration with existing tools like Prometheus or Grafana. If any performance degradation or model drift is detected, the system triggers alerts, enabling quick intervention by cross-functional teams. The system also supports model retraining by feeding new data into the workflow and updating models accordingly. These updates are then logged in the registry, ensuring that the model's lifecycle is well-documented and that any changes are tracked.

3.3 Integration with Tech Teams

A key strength of the proposed system is its ability to facilitate seamless collaboration between cross-functional teams. By offering a unified platform that integrates with various tools used by different teams, the system bridges the gap between data scientists, engineers, and business stakeholders [71]. Data scientists benefit from the model registry by being able to track and manage the versions of their models while experimenting with different techniques. They can also share their models with engineers and business teams for feedback and iteration, fostering a more collaborative approach to AI model development [17, 35].

Engineers, who are responsible for deploying and scaling AI models, can use the registry's version control and metadata management features to identify the correct models for production environments. They can also collaborate with data scientists by providing feedback on model performance and integration issues, ensuring that the models are optimized for real-world use. The system's integration with deployment tools like Kubernetes and Jenkins further streamlines the collaboration process, allowing engineers to automate the deployment and monitoring of models with minimal manual intervention [72, 73].

Business stakeholders, who are often more removed from the technical aspects of AI development, can use the system to track the performance of deployed models and assess their impact on key business metrics. The registry's transparent version control and detailed metadata provide business users with the information needed to make data-driven decisions, ensuring that AI models align with organizational goals [17, 71]. Additionally, the system supports communication features, such as automatic notifications and task management, that keep all teams informed and aligned on the status of different models throughout their lifecycle. This level of integration ensures that cross-functional collaboration is not only possible but also streamlined, resulting in a more efficient and effective AI model management process [35, 36, 40].

Implementation Strategy

4.1 System Development and Tools

The development of the AI model registry and lifecycle management system will follow a modular approach to ensure flexibility and scalability. The core of the system will be built using microservices architecture, which will allow different components—such as version control, metadata management, and deployment—to function independently but cohesively [74, 75]. For backend development, Python is an ideal language due to its robust ecosystem for machine learning and data science, including libraries like TensorFlow, PyTorch, and Scikit-learn. Additionally, Flask or FastAPI will be used to build lightweight RESTful APIs that facilitate communication between the registry and external tools [76, 77].

For database management, a relational database such as PostgreSQL will be employed to store model metadata, including version information, model performance metrics, and deployment status. This ensures that all data is structured and easily accessible for querying and reporting. In addition, cloud storage solutions like Amazon S3 or Google Cloud Storage will be leveraged to handle the storage of large model files, ensuring that models are stored securely and are accessible at any time [78, 79].

The frontend of the system will be built using React, which will provide a user-friendly interface for different stakeholders to interact with the registry. This will allow users to visualize model metadata, browse through model versions, and initiate deployment workflows with ease. To support automation and scalability, Kubernetes will be used for container orchestration, ensuring that the system can handle increasing loads as the number of models and users grows. Continuous integration and deployment (CI/CD) pipelines will be implemented using Jenkins or GitLab CI to ensure that updates to the system are seamlessly integrated and deployed to production without disrupting ongoing operations [80].

4.2 Collaboration Tools

Integrating collaboration tools like GitHub, Jira, and Slack will significantly enhance communication and collaboration across the various teams involved in the AI model lifecycle. GitHub will be central to version control for code, with integration into the model registry system to track changes made to both code and models [81, 82]. Each update to the model repository can trigger automatic versioning within the registry, ensuring that data scientists, engineers, and other stakeholders are always working with the most up-to-date models. This integration will minimize the risk of version conflicts and streamline workflows between data scientists and engineers by automating the flow of information between code and model updates [80, 83].

Jira will be used for project management and task tracking. Tasks related to AI model development, deployment, and monitoring will be created and tracked within Jira, with each task linked to the corresponding model version and metadata within the registry. This will ensure that all teams are aware of the status of ongoing tasks and any dependencies between them [84, 85]. For example, if a data scientist updates a model, a corresponding Jira ticket can be created to notify engineers that a new version is available for testing and deployment. Jira's integration with GitHub will allow for automated linking of code changes to project tasks, ensuring that there is a clear audit trail of actions taken throughout the model's lifecycle [85, 86].

Slack, as a communication tool, will be used to facilitate real-time communication among cross-functional teams. The registry system will integrate with Slack to send notifications about model updates, deployment status, or performance issues [87, 88]. This ensures that everyone involved in the process is immediately informed about important events, reducing delays caused by missed communications. Moreover, Slack channels can be set up for specific model projects, where data scientists, engineers, and business stakeholders can collaborate and discuss challenges, results, and next steps in a focused and organized manner [89, 90].

4.3 Deployment and Monitoring

Deployment and monitoring are crucial for ensuring the long-term success of AI models, and the proposed system will adopt a hybrid approach to both. The deployment of AI models will be managed using containerization technologies such as Docker and orchestration tools like Kubernetes [21]. Docker allows models to be packaged along with their dependencies, ensuring consistency across different environments, from development to production. Kubernetes will provide the orchestration layer to automate the deployment, scaling, and management of these containers, ensuring that AI models are deployed efficiently and can scale automatically based on demand [91, 92].

The deployment process will be integrated with CI/CD pipelines to streamline updates and ensure that the latest model versions are deployed with minimal manual intervention [93-95]. When a new model is ready for deployment, the registry will trigger an automatic pipeline in Jenkins or GitLab CI to build, test, and deploy the model to the appropriate environment. The system will also allow for automated rollbacks to previous model versions if issues arise during deployment, ensuring minimal disruption to production systems [94].

For monitoring, real-time performance tracking will be implemented using tools like Prometheus and Grafana. These tools will collect and visualize key performance metrics such as model accuracy, latency, and resource usage [96]. Any deviations from predefined thresholds—such as a sudden drop in accuracy or increased latency—will trigger alerts to notify relevant stakeholders [97, 98]. Additionally, the system will support model drift detection, which automatically flags changes in data distribution that may affect model performance [99, 100]. If drift is detected, the system will suggest retraining the model with fresh data, ensuring that models remain accurate and reliable over time. This integrated approach to deployment and monitoring will help ensure that AI models continue to operate effectively and that any issues are quickly identified and addressed [95, 98].

Conclusion and Recommendations

5.1 Conclusion

This paper has presented a comprehensive approach to developing an AI model registry and lifecycle management system, focusing on addressing the key challenges faced by cross-functional teams in managing AI models. The proposed system is designed to streamline model versioning, metadata management, and deployment processes, offering features like role-based access control, automated version tracking, and seamless integration with existing machine learning platforms and tools. By offering a centralized model registry, this system ensures that models are easily accessible, well-documented, and secure throughout their lifecycle, from development to deployment and monitoring.

One of the primary contributions of this work is the introduction of an integrated solution that supports collaboration between data scientists, engineers, and business stakeholders. Through the integration of tools like GitHub, Jira, and Slack, the system fosters real-time communication and project management, allowing teams to collaborate more efficiently and effectively. Additionally, the lifecycle management workflow proposed in this paper ensures that models are continuously monitored, and performance issues can be detected and addressed promptly, enhancing model reliability and reducing the risk of deployment failures. The system's use of containerization and cloud-native infrastructure also ensures scalability, making it suitable for organizations of various sizes.

The paper also highlights the importance of a robust system for managing AI models, especially as AI continues to evolve and become more embedded across different industries. The model registry and lifecycle management system presented in this paper is not only a technical contribution but also a strategic one, offering organizations a way to optimize their AI operations, ensure model governance, and facilitate collaboration among diverse teams. This work lays the foundation for more efficient AI development and deployment practices, setting the stage for future advancements in the field.

The implementation of the proposed AI model registry and lifecycle management system has profound implications for cross-functional teams involved in AI model development and deployment. One of the most significant benefits is the improvement in efficiency. By centralizing model versioning and metadata management, teams can quickly access the information they need without wasting time searching through different repositories or tracking down specific versions. Automated processes for deployment, rollback, and monitoring ensure that teams can focus more on high-value tasks such as model optimization and strategy, rather than dealing with manual or repetitive processes.

The system also enhances collaboration between data scientists, engineers, and business stakeholders. In traditional AI development environments, communication breakdowns often occur due to the siloed nature of different teams. Data scientists may develop models without a clear understanding of deployment constraints, while engineers may not be fully aware of the underlying model assumptions. By integrating project management tools like Jira and GitHub, as well as communication platforms like Slack, the system ensures that all teams stay aligned, share information in real time, and provide feedback at each stage of the model's lifecycle. This shared understanding leads to more informed decision-making, faster iteration, and ultimately, more successful AI deployments.

Moreover, the proposed system helps improve the quality of AI models. Continuous monitoring, automatic model drift detection, and the ability to quickly roll back to previous versions if issues arise allow teams to maintain model performance over time. This ensures that models remain accurate, relevant, and robust, even as data changes or new technologies emerge. By fostering collaboration and providing a transparent, efficient process for model management, the system enhances both the speed and quality of AI development, leading to better outcomes for organizations and their stakeholders.

5.2 Future Research Directions

Although the proposed system offers a comprehensive solution for managing AI models, there are several areas where further research can expand and refine the framework. One potential area for future exploration is the automation of model updates. While the system currently allows for the manual intervention of model deployment and monitoring, research into fully automated model retraining and updating based on real-time data could further streamline the workflow. This would reduce the need for human intervention, allowing models to adapt to changes in data patterns more quickly and efficiently. Automation in this regard could be particularly valuable in dynamic environments where data is continuously evolving.

Another area for future research is the scalability of the system, particularly in large, complex organizations. As AI models become more integrated into various business units and applications, the need for scalable solutions that can handle large volumes of models, users, and data becomes critical. Research into distributed systems and architectures that can scale horizontally while maintaining performance and security would be valuable in addressing the challenges faced by large enterprises. Furthermore, investigating the role of edge computing in AI model deployment could open new possibilities for real-time model updates and monitoring in resource-constrained environments.

Lastly, future research could focus on improving the interoperability of the system with other tools and platforms. While this paper discusses integration with GitHub, Jira, and Slack, there are many other tools and frameworks used in AI development, including TensorFlow Extended (TFX), Apache Airflow, and Databricks. Exploring how the registry and lifecycle management system can integrate with these platforms could enhance its utility and adoption in diverse organizational settings. Research into the development of open standards and protocols for AI model management could also play a key role in fostering industry-wide best practices and enabling better cross-platform integration.

REFERENCES

[1]. M. Chui and S. Francisco, "Artificial intelligence the next digital frontier," McKinsey and Company Global Institute, vol. 47, no. 3.6, pp. 6-8, 2017.

[2]. J. Bughin, E. Hazan, P. Sree Ramaswamy, W. DC, and M. Chu, "Artificial intelligence the next digital frontier," 2017.

[3]. S. GÜNDÜZ, "A Comparative Research on Artificial Intelligence-Driven Transformations in Business Management: Strategic Applications in Finance, Tourism, Healthcare, Retail, and Manufacturing Sectors," ULUSLARARASI AKADEMİK BİRİKİM DERGİSİ, vol. 7, no. 4, 2024.

[4]. P. Agarwal, S. Swami, and S. K. Malhotra, "Artificial intelligence adoption in the post COVID-19 new-normal and role of smart technologies in transforming business: a review," Journal of Science and Technology Policy Management, vol. 15, no. 3, pp. 506-529, 2024.

[5]. M. Javaid, A. Haleem, R. P. Singh, and R. Suman, "Artificial intelligence applications for industry 4.0: A literature-based study," Journal of Industrial Integration and Management, vol. 7, no. 01, pp. 83-111, 2022.

[6]. A. B. Rashid and A. K. Kausik, "AI revolutionizing industries worldwide: A comprehensive overview of its diverse applications," Hybrid Advances, p. 100277, 2024.

[7]. S. Abbasi and A. M. Rahmani, "Artificial intelligence and software modeling approaches in autonomous vehicles for safety management: a systematic review," Information, vol. 14, no. 10, p. 555, 2023.

[8]. I. H. Sarker, "AI-based modeling: techniques, applications and research issues towards automation, intelligent and smart systems," SN computer science, vol. 3, no. 2, p. 158, 2022.

[9]. Y. Gil et al., "Artificial intelligence for modeling complex systems: taming the complexity of expert models to improve decision making," ACM Transactions on Interactive Intelligent Systems, vol. 11, no. 2, pp. 1-49, 2021.

[10]. G. Dagnaw, "Artificial intelligence towards future industrial opportunities and challenges," 2020.

[11]. O. Michael, "of Artificial Intelligence," The Future of Small Business in Industry 5.0, p. 215, 2024.

[12]. M. Kejriwal, Artificial intelligence for industries of the future. Springer, 2023.

[13]. Y. Duan, J. S. Edwards, and Y. K. Dwivedi, "Artificial intelligence for decision making in the era of Big Data–evolution, challenges and research agenda," International journal of information management, vol. 48, pp. 63-71, 2019.

[14]. M. F. Tahir, "Role of AI/ML in decision making in software release management," 2024.

[15]. A. Soni, A. Kumar, R. Arora, and R. Garine, "Integrating AI into the Software Development Life Cycle: Best Practices, Tools, and Impact Analysis," Tools, and Impact Analysis (June 10, 2023), 2023.

[16]. W. Hummer et al., "Modelops: Cloud-based lifecycle management for reliable and trusted ai," in 2019 IEEE International Conference on Cloud Engineering (IC2E), 2019: IEEE, pp. 113-120.

[17]. R. U. Attah, B. M. P. Garba, I. Gil-Ozoudeh, and O. Iwuanyanwu, "Cross-functional team dynamics in technology management: a comprehensive review of efficiency and innovation enhancement," Eng Sci Technol J, vol. 5, no. 12, pp. 3248-65, 2024.

[18]. O. G. Berg, "Challenges and success factors in a cross-functional development team in a large-scale agile context: An exploratory case study," NTNU, 2024.

[19]. S. Schelter, F. Biessmann, T. Januschowski, D. Salinas, S. Seufert, and G. Szarvas, "On challenges in machine learning model management," 2015.

[20]. D. De Silva and D. Alahakoon, "An artificial intelligence life cycle: From conception to production," Patterns, vol. 3, no. 6, 2022.

[21]. S. Sinha and Y. M. Lee, "Challenges with developing and deploying AI models and applications in industrial systems," Discover Artificial Intelligence, vol. 4, no. 1, p. 55, 2024.

[22]. S. Passi and S. J. Jackson, "Trust in data science: Collaboration, translation, and accountability in corporate data science projects," Proceedings of the ACM on human-computer interaction, vol. 2, no. CSCW, pp. 1-28, 2018.

[23]. A. X. Zhang, M. Muller, and D. Wang, "How do data science workers collaborate? roles, workflows, and tools," Proceedings of the ACM on Human-Computer Interaction, vol. 4, no. CSCW1, pp. 1-23, 2020.

[24]. R. Avacharmal and S. Pamulaparthyvenkata, "Enhancing Algorithmic Efficacy: A Comprehensive Exploration of Machine Learning Model Lifecycle Management from Inception to Operationalization," Distributed Learning and Broad Applications in Scientific Research, vol. 8, pp. 29-45, 2022.

[25]. M. Elahi, S. O. Afolaranmi, J. L. Martinez Lastra, and J. A. Perez Garcia, "A comprehensive literature review of the applications of AI techniques through the lifecycle of industrial equipment," Discover Artificial Intelligence, vol. 3, no. 1, p. 43, 2023.

[26]. R. Ashmore, R. Calinescu, and C. Paterson, "Assuring the machine learning lifecycle: Desiderata, methods, and challenges," ACM Computing Surveys (CSUR), vol. 54, no. 5, pp. 1-39, 2021.

[27]. D. Nigenda et al., "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, 2022, pp. 3671-3681.

[28]. L. Yang and D. Rossi, "Quality monitoring and assessment of deployed deep learning models for network AIOps," IEEE Network, vol. 35, no. 6, pp. 84-90, 2022.

[29]. S. Patchipala, "Tackling data and model drift in AI: Strategies for maintaining accuracy during ML model inference," International Journal of Science and Research Archive, vol. 10, no. 2, pp. 1198-1209, 2023.

[30]. F. Bachinger, G. Kronberger, and M. Affenzeller, "Continuous improvement and adaptation of predictive models in smart manufacturing and model management," IET Collaborative Intelligent Manufacturing, vol. 3, no. 1, pp. 48-63, 2021.

[31]. M. Yusuff, "Model Drift Monitoring: Continuously Tracking Model Performance Metrics to Detect Accuracy Degradation."

[32]. F. Bayram, B. S. Ahmed, and A. Kassler, "From concept drift to model degradation: An overview on performance-aware drift detectors," Knowledge-Based Systems, vol. 245, p. 108632, 2022.

[33]. K. Dubovikov, Managing Data Science: Effective strategies to manage data science projects and build a sustainable team. Packt Publishing Ltd, 2019.

[34]. L. E. Lwakatare, A. Raj, I. Crnkovic, J. Bosch, and H. H. Olsson, "Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions," Information and software technology, vol. 127, p. 106368, 2020.

[35]. A. Goyal, "Enhancing engineering project efficiency through cross-functional collaboration and IoT integration," Int. J. Res. Anal. Rev, vol. 8, no. 4, pp. 396-402, 2021.

[36]. T. Ahmad, J. Boit, and A. Aakula, "The role of cross-functional collaboration in digital transformation," Journal of Computational Intelligence and Robotics, vol. 3, no. 1, pp. 205-42, 2023.

[37]. S. U. Rahaman, "Breaking Silos: Architecting Cross-Functional Analytics Frameworks for Collaborative Insights."

[38]. W. H. Deng, N. Yildirim, M. Chang, M. Eslami, K. Holstein, and M. Madaio, "Investigating practices and opportunities for cross-functional collaboration around AI fairness in industry practice," in Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency, 2023, pp. 705-716.

[39]. N. L. Rane, "Multidisciplinary collaboration: key players in successful implementation of ChatGPT and similar generative artificial intelligence in manufacturing, finance, retail, transportation, and construction industry," 2023.

[40]. B. Saha and M. Kumar, "Investigating cross-functional collaboration and knowledge sharing in cloudnative program management systems," International Journal for Research in Management and Pharmacy, vol. 9, no. 12, 2020.

[41]. P. Goncharenko, "The Potential of Using Artificial Intelligence in Communications within Project Management," 2024.

[42]. F. Imran, K. Shahzad, A. Butt, and J. Kantola, "Digital transformation of industrial organizations: Toward an integrated framework," Journal of change management, vol. 21, no. 4, pp. 451-479, 2021.

[43]. M. Pöyhönen, "Human-AI Integration in Long-Established Organizations," 2024.

[44]. O. H. Olayinka, "Data driven customer segmentation and personalization strategies in modern business intelligence frameworks," World Journal of Advanced Research and Reviews, vol. 12, no. 3, pp. 711-726, 2021.

[45]. J. Jöhnk, M. Weißert, and K. Wyrtki, "Ready or not, AI comes—an interview study of

organizational AI readiness factors," Business & information systems engineering, vol. 63, no. 1, pp. 5-20, 2021.

[46]. R. Bergmann, F. Theusch, P. Heisterkamp, and N. Grigoryan, "Comparative Analysis of Open-Source ML Pipeline Orchestration Platforms."

[47]. R. Kienzler and H. Kyas, "Tensorflow 2.0 and Kubeflow for Scalable and Reproducable Enterprise AI," in CS & IT Conference Proceedings, 2020, vol. 10, no. 1: CS & IT Conference Proceedings.

[48]. A. Choudhury, Continuous Machine Learning with Kubeflow: Performing Reliable MLOps with Capabilities of TFX, Sagemaker and Kubernetes (English Edition). BPB Publications, 2021.

[49]. D. G. Ancona and D. F. Caldwell, "Information technology and work groups: The case of new product teams," in Intellectual teamwork: Psychology Press, 2014, pp. 173-190.

[50]. J. Marttila, "ENHANCING MULTI-PROJECT MANAGEMENT THROUGH PRODUCT MANAGEMENT INTEGRATION," 2024.

[51]. R. Y. Gabrow, "Concurrent engineering, product life cycle management using cross-functional teams: a case study," Periodicals of Engineering and Natural Sciences (PEN), vol. 9, no. 2, pp. 842-857, 2021.

[52]. M. Shahin, M. A. Babar, and L. Zhu, "Continuous integration, delivery and deployment: a systematic review on approaches, tools, challenges and practices," IEEE access, vol. 5, pp. 3909-3943, 2017.

[53]. W. Shen et al., "Systems integration and collaboration in architecture, engineering, construction, and facilities management: A review," Advanced engineering informatics, vol. 24, no. 2, pp. 196-207, 2010.

[54]. R. Van Der Lans, Data Virtualization for business intelligence systems: revolutionizing data integration for data warehouses. Elsevier, 2012.

[55]. A. Shrivastav, "THE ROLE OF AGILE METHODOLOGIES IN PRODUCT LIFECYCLE MANAGEMENT (PLM) OPTIMIZATION."

[56]. M. J. Goswami, "Optimizing Product Lifecycle Management with AI: From Development to Deployment," International Journal of Business Management and Visuals, ISSN, pp. 3006-2705.

[57]. E. Papagiannidis, I. M. Enholm, C. Dremel, P. Mikalef, and J. Krogstie, "Toward AI governance: Identifying best practices and potential barriers and outcomes," Information Systems Frontiers, vol. 25, no. 1, pp. 123-141, 2023.

[58]. M. Dumas et al., "AI-augmented business process management systems: a research manifesto," ACM Transactions on Management Information Systems, vol. 14, no. 1, pp. 1-19, 2023.

[59]. H. Lakk, "Model-driven role-based access control for databases," Citeseer, 2012.

[60]. M. Uddin, S. Islam, and A. Al-Nemrat, "A dynamic access control model using authorising workflow and task-role-based access control," Ieee Access, vol. 7, pp. 166676-166689, 2019.

[61]. T. Baumer, M. Müller, and G. Pernul, "System for cross-domain identity management (SCIM): Survey and enhancement with RBAC," IEEE Access, vol. 11, pp. 86872-86894, 2023.

[62]. A. Alabdulatif, N. N. Thilakarathne, and K. Kalinaki, "A novel cloud enabled access control model for preserving the security and privacy of medical big data," Electronics, vol. 12, no. 12, p. 2646, 2023.

[63]. S. O. Dyke et al., "Registered access: authorizing data access," European Journal of Human Genetics, vol. 26, no. 12, pp. 1721-1731, 2018.

[64]. A. Petrova, "Cloud Computing in the Age of Big Data: Storage, Analytics, and Scalability," Advances in Computer Sciences, vol. 6, no. 1, 2023.

[65]. D. Bayazitov, K. Kozhakhmet, A. Omirali, and R. Zhumaliyeva, "Leveraging Amazon Web Services for cloud storage and AI algorithm integration: A comprehensive analysis," Applied Mathematics, vol. 18, no. 6, pp. 1235-1246, 2024.

[66]. P. Mathur, "Cloud computing infrastructure, platforms, and software for scientific research," High Performance Computing in Biomimetics: Modeling, Architecture and Applications, pp. 89-127, 2024.

[67]. M. Haakman, L. Cruz, H. Huijgens, and A. Van Deursen, "AI lifecycle models need to be revised: An exploratory study in Fintech," Empirical Software Engineering, vol. 26, no. 5, p. 95, 2021.

[68]. S. Yajamanam Kidambi, "End-to-End artificial intelligence lifecycle management," Massachusetts Institute of Technology, 2022.

[69]. Y. Xie, "AI Model Lifecycle Management: Systematic Mapping Study and Solution for AI Democratisation," 2020.

[70]. A. K. Kordon and A. K. Kordon, "The AI-based data science workflow," Applying Data Science: How to Create Value with Artificial Intelligence, pp. 189-202, 2020.

[71]. E. A. d. Oliveira, M. L. Pimenta, P. Hilletofth, and D. Eriksson, "Integration through cross-functional teams in a service company," European Business Review, vol. 28, no. 4, pp. 405-430, 2016.

[72]. G. Abdiyeva-Aliyeva, "Application of ai in software engineering: Handling data management problems in production," Informatics and Control Problems, vol. 43, no. 2, pp. 94-101, 2023.

[73]. J. Paul, "How Software Engineering is Shaping AI's Future: The Tools and Practices Behind Smarter Systems," 2024.

[74]. D. K. Pandiya and N. Charankar, "INTEGRATION OF MICROSERVICES AND AI FOR REAL-TIME DATA PROCESSING."

[75]. E. Oye, E. Frank, and J. Owen, "Microservices Architecture for Large-Scale AI Applications," 2024.

[76]. D. V. A. Palli, "Monolithic vs. Microservices Architectures for AI-Integrated Applications," 2024.

[77]. E. Wolff, Microservices: flexible software architecture. Addison-Wesley Professional, 2016.

[78]. M. Abbasi, M. V. Bernardo, P. Váz, J. Silva, and P. Martins, "Adaptive and Scalable Database Management with Machine Learning Integration: A PostgreSQL Case Study," Information, vol. 15, no. 9, p. 574, 2024.

[79]. W. L. Schulz, B. G. Nelson, D. K. Felker, T. J. Durant, and R. Torres, "Evaluation of relational and NoSQL database architectures to manage genomic annotations," Journal of biomedical informatics, vol. 64, pp. 288-295, 2016.

[80]. O. Ur Rehman, "Frontend Module for the Management of Museums and Events," Politecnico di Torino, 2024.

[81]. H. Tuunainen, "The potential of generative artificial intelligence in leading a scalable agile enterprise by objectives," 2024.

[82]. H. S. Vudandapuram and R. R. Madireddy, "Evaluating Requirement Management Tool's Features for Effective Collaboration in Distributed Software Development Teams," ed, 2024.

[83]. V. Ståhlberg, "Enhancing software development processes with artificial intelligence," Artificial intelligence (AI), 2024.

[84]. F. El Aouni, K. Moumane, S. Bendaouya, A. Ait Laasri, and Y. Wahi, "Development of a JIRA Plugin for Tracking and Managing Time Allocations for Tasks and Projects," in 2024 World Conference on Complex Systems (WCCS), 2024: IEEE, pp. 1-7.

[85]. O. AlHarbi, R. AlMalki, and N. AlYousef, "Advancing Project Management Methodologies: An In-Depth Analysis of Jira in

Managerial and Developmental Contexts," International Journal of Technology Innovation and Management (IJTIM), vol. 3, no. 2, pp. 40-59, 2023.

[86]. S. R. Keshireddy, "AI Driven Strategies for Efficient Project Tracking and Delivery in Software Engineering Management," Research Briefs on Information and Communication Technology Evolution, vol. 9, pp. 228-249, 2023.

[87]. E. Kalelioğlu, Implementing Atlassian Confluence: Strategies, tips, and insights to enhance distributed team collaboration using Confluence. Packt Publishing Ltd, 2023.

[88]. E. Stoeckli, C. Dremel, F. Uebernickel, and W. Brenner, "How affordances of chatbots cross the chasm between social and traditional enterprise systems," Electronic Markets, vol. 30, pp. 369-403, 2020.

[89]. M. S. Farooq, Z. Kalim, J. N. Qureshi, S. Rasheed, and A. Abid, "A blockchain-based framework for distributed agile software development," IEEe Access, vol. 10, pp. 17977-17995, 2022.

[90]. B. Familiar and J. Barnes, "Business in Real-Time Using Azure IoT and Cortana Intelligence Suite," Apress: Berkeley, CA, USA, 2017.

[91]. B. Saha, "Evaluating the impact of AI-driven project prioritization on program success in hybrid cloud environments," Available at SSRN 5224739, 2019.

[92]. Z. Nishtar and J. Afzal, "A Review of real-time monitoring of hybrid energy systems by using artificial intelligence and IoT," Pakistan Journal of Engineering and Technology, vol. 6, no. 3, pp. 8-15, 2023.

[93]. V. U. Ugwueze and J. N. Chukwunweike, "Continuous integration and deployment strategies for streamlined DevOps in software engineering and application delivery," Int J Comput Appl Technol Res, vol. 14, no. 1, pp. 1-24, 2024.

[94]. P. S. Chatterjee and H. K. Mittal, "Enhancing Operational Efficiency through the Integration of CI/CD and DevOps in Software Deployment," in 2024 Sixth International Conference on Computational Intelligence and Communication Technologies (CCICT), 2024: IEEE, pp. 173-182.

[95]. A. MUSTYALA, "CI/CD Pipelines in Kubernetes: Accelerating Software Development and Deployment," EPH-International Journal of Science And Engineering, vol. 8, no. 3, pp. 1-11, 2022.

[96]. N. Dimonte, "Centralized Monitoring Infrastructure on Cloud: An Open Source Approach," Politecnico di Torino, 2024.

[97]. T. J. Akinbolaji, G. Nzeako, D. Akokodaripon, and A. V. Aderoju, "Proactive monitoring and security in cloud infrastructure: Leveraging tools like Prometheus, Grafana, and HashiCorp Vault for robust DevOps practices," World Journal of Advanced Engineering Technology and Sciences, vol. 13, no. 2, pp. 90-104, 2024.

[98]. P. Chintale, DevOps Design Pattern: Implementing DevOps best practices for secure and reliable CI/CD pipeline (English Edition). Bpb Publications, 2023.

[99]. V. V. R. Boda and J. Immaneni, "Optimizing CI/CD in Healthcare: Tried and True Techniques," International Journal of Emerging Research in Engineering and Technology, vol. 3, no. 2, pp. 28-38, 2022.

[100]. S. Banala, "DevOps Essentials: Key Practices for Continuous Integration and Continuous Delivery," International Numeric Journal of Machine Learning and Robots, vol. 8, no. 8, pp. 1-14, 2024.