

# Industrial MLOps: a systematic review of architectures and implementation challenges

Mohan Rajashekarappa , Ebru Turanoglu Bekar , Alexander Karlsson , Jon Bokrantz , Mukund Subramanian & Anders Skoogh

To cite this article: Mohan Rajashekarappa , Ebru Turanoglu Bekar , Alexander Karlsson , Jon Bokrantz , Mukund Subramanian & Anders Skoogh (2026) Industrial MLOps: a systematic review of architectures and implementation challenges, Production & Manufacturing Research, 14:1, 2658878, DOI: [10.1080/21693277.2026.2658878](https://doi.org/10.1080/21693277.2026.2658878)

To link to this article: <https://doi.org/10.1080/21693277.2026.2658878>



© 2026 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group.



Published online: 14 Apr 2026.



Submit your article to this journal [↗](#)



Article views: 183



View related articles [↗](#)



View Crossmark data [↗](#)

# Industrial MLOps: a systematic review of architectures and implementation challenges

Mohan Rajashekarappa<sup>a</sup>, Ebru Turanoglu Bekar<sup>a</sup>, Alexander Karlsson<sup>b</sup>, Jon Bokrantz<sup>a</sup>, Mukund Subramaniyan<sup>c</sup> and Anders Skoogh<sup>a</sup>

<sup>a</sup>Department of Mechanical Engineering, Chalmers University of Technology, Gothenburg, Sweden; <sup>b</sup>School of Informatics, University of Skövde, Skövde, Sweden; <sup>c</sup>Innovation Lab, Supply Chain Digital, Volvo Cars Corporation, Gothenburg, Sweden

## ABSTRACT

The rise of advanced digitalization in Industry 4.0 has enabled manufacturers to leverage data through AI and ML solutions for various manufacturing challenges. However, integrating these models into factory settings remains challenging, as models that perform well on static datasets struggle with dynamic shop floor data. MLOps is an emerging discipline focused on bridging the gap between ML models and production environments; however, in the manufacturing domain, questions remain about how to effectively deploy ML models using MLOps. This article addresses these gaps by conducting a systematic literature review combined with thematic analysis to explore architectures and frameworks used to adopt MLOps in real-world industrial applications, referred to here as industrial MLOps. The study identifies key architectural requirements and outlines seven implementation challenges, with recommendations and architecture mappings to overcome them. Results show that fully automated MLOps frameworks remain underdeveloped, and that modular, scalable architectures are recommended to address model drift, data quality, and integration challenges.

## ARTICLE HISTORY

Received 22 May 2025  
Accepted 8 April 2026



## KEYWORDS

Machine learning operations (MLOps); machine learning (ML); artificial intelligence (AI); deployment challenges; systematic literature review (SLR)

## 1. Introduction

Artificial intelligence (AI) and machine learning (ML) solutions have been playing an important role in enhancing data-driven decision-making, steering towards intelligent manufacturing processes. Implementation and novelty are the characteristics of production innovation contributing to the success of the initiative (Larsson, 2017). It is evident that manufacturing organizations have perceived AI and ML as innovative and promising technological solutions to enhance their manufacturing processes. It is important that innovative technology is implemented and adds value to the organization. However, scientific literature highlights that the emphasis has predominantly been on developing and evaluating AI/ML models rather than on deployment, which faces considerable challenges (Schröer et al., 2021). As a result, a significant portion of proof-of-concept models never advance to production (Kreuzberger et al., 2023). The adoption of AI/ML technology is posing a challenge in the context of manufacturing, which needs to be addressed.

Machine learning operations, in short, MLOps, is a developing discipline that connects ML models with production environments. Its goal is to enhance the deployment, monitoring, and upkeep of ML models, making these processes more scalable and methodical. MLOps is an interdisciplinary approach that combines ML, software engineering, and data engineering practices to enhance the automation and reproducibility of ML models in production environments (Kreuzberger et al., 2023). The concept is built on the foundational principles of development operations (DevOps), focusing on the continuous integration and deployment (CI/CD) of ML systems to ensure robust, efficient, and error-free operations. MLOps adapts DevOps principles to specifically address the unique challenges encountered throughout the lifecycle of ML models. This includes aspects such as model versioning, ensuring data quality, and

**CONTACT** Mohan Rajashekarappa  [rmohan@chalmers.se](mailto:rmohan@chalmers.se)  Department of Mechanical Engineering, Chalmers University of Technology, Hörsalsvägen 7A SE-412 96, Gothenburg, Sweden

© 2026 The Author(s). Published by Informa UK Limited, trading as Taylor & Francis Group. This is an Open Access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited. The terms on which this article has been published allow the posting of the Accepted Manuscript in a repository by the author(s) or with their consent.

automating the training and testing of models. Successful projects in the field of AI/ML require strong technological capabilities and solutions like MLOps, which play a vital role in enabling companies to leverage AI for decision-making.

In this context, while MLOps is essential for utilizing AI to stay competitive in technological innovation, this article acknowledges the challenges involved in implementing, automating, and operationalizing AI and ML technologies within the shop floor. This article aims to bridge this gap by utilizing systematic literature review (SLR) methodology to highlight the architectures, frameworks, and challenges documented in scientific literature, offering insights into how MLOps methodologies can be leveraged to successfully operationalize ML models in real manufacturing environments. The study looks into scientific articles that apply MLOps to industrial use cases, discuss the challenges encountered during MLOps implementation in industry and propose unique architectures and frameworks for adopting MLOps. The article also reflects on the common themes among the identified architectures to uncover the foundational elements necessary for the successful implementation of ML in an industrial context. The research questions addressed in this article are as follows:

- 1) What are the key features of MLOps architectures that contribute to the deployment of ML models specific to manufacturing?
- 2) What are the challenges faced during the implementation of ML models in manufacturing using MLOps, and how have they been addressed?

This study makes unique contributions to the MLOps literature by addressing a critical gap in manufacturing-specific implementations. While existing reviews have extensively documented general MLOps principles, tools, and challenges across enterprise contexts, and recent work has begun exploring Industry 4.0 settings through observational studies and company interviews, there remains a need for systematic reviews that focus specifically on empirically grounded MLOps architectures with real manufacturing deployment evidence. This study goes beyond theoretical frameworks by conducting a systematic literature review combined with thematic analysis to examine MLOps architectures that have either been implemented with actual production data or demonstrate clear implementation readiness for manufacturing environments. Unlike prior work that treats MLOps and manufacturing challenges separately, our research investigates the unique convergence challenges that emerge when operationalizing ML models in industrial settings. Furthermore, this study provides a novel challenge-architecture mapping framework that systematically connects identified implementation barriers to specific architectural solutions, offering practitioners evidence-based pathways for MLOps adoption tailored to manufacturing contexts. This implementation-focused approach, emphasizing both architectural characteristics and practical deployment validation, and the architecture network mapping, distinguishes our work from existing literature.

[Section 2](#) provides a background and motivation for this study, which is followed by [Section 3](#), which provides details regarding the SLR methodology adopted for the study. [Section 4](#) describes in detail the results obtained from this study and these results are discussed in [Section 5](#). The article is concluded in [Section 6](#) with theoretical implications and future work.

## 2. Background

### 2.1. ML model deployment challenges

Deploying ML models in manufacturing involves numerous challenges in both technical and organizational aspects. Although ML/AI has shown remarkable promise in improving production processes through predictive maintenance, quality control, and optimization, the real world is different from controlled experimental conditions. This is where the true hurdles of deployment emerge. One of the primary issues is the development–deployment discrepancy, where models trained in controlled environments often fail to perform optimally in production due to real-world complexities. This challenge is discussed by Andrei Paleyes (Lawrence et al., 2022), who emphasizes that the transition from development to deployment is full of uncertainties that compromise model performance in production environments (Lawrence et al., 2022). Technological challenges further complicate the deployment process. The rapidly

evolving software tools for AI applications make it difficult to ensure that the models remain compatible with production systems (Heymann et al., 2023). Moreover, these challenges are not limited to technology alone. Organizational barriers, such as the reluctance of process experts to trust ML systems, play an important role in deployment failures. For successful deployment, gaining stakeholder trust and aligning ML models with business objectives are very important factors (Heymann et al., 2022).

Another significant hurdle is related to data availability and quality. Data collection and preprocessing consume a significant portion of resources in ML projects. Without high-quality, labeled data, model performance suffers, making deployment in manufacturing environments particularly challenging (Mayr et al., 2019). In addition, MLOps, which offers a potential solution, requires substantial changes in organizational infrastructure to achieve continuous integration and deployment (Subramanya et al., 2022). Amou Najafabadi (2024) highlights three key challenges in MLOps: First, there is no consensus on the optimal design and implementation of MLOps workflows. Second, while many tools exist for different stages of MLOps, there are no clear guidelines on how to integrate them into a complete system. Third, the balance between human involvement and automation in MLOps remains unclear.

## 2.2. MLOps as an innovative approach in production environments

MLOps is an engineering discipline focused on managing the entire lifecycle of machine learning products, from their initial development to deployment, monitoring, and scaling. It combines best practices from machine learning, software engineering (specifically DevOps), and data engineering to streamline the transition of ML systems from development into production environments. MLOps builds on the foundation of DevOps, which integrates software development and IT operations to shorten the development lifecycle and ensure frequent, business-aligned updates. DevOps promotes collaboration between development and operations teams through continuous integration, continuous delivery, and automation (Subramanya et al., 2022). The principles of MLOps emphasize improvements in efficiency, adaptability, and overall system effectiveness. MLOps introduces streamlined processes such as automated CI/CD, clear task management with directed acyclic graphs, and detailed tracking of metadata. These practices transform the traditional management of machine learning models by making the processes faster, more reliable, and easier to manage. By ensuring continuous training and rigorous performance monitoring, MLOps keeps models relevant and effective, making systems better adapted to meet both current and future demands. Core principles of MLOps include (Kreuzberger et al., 2023):

*Automation through CI/CD:* this system automatically manages the build, test, delivery, and deployment phases, providing quick feedback on operations to implement core DevOps strategies.

*Experiment replicability:* ensuring that any ML experiment can be duplicated with identical results is critical for maintaining scientific validity. This principle emphasizes the ability to consistently replicate the outcomes of experiments.

*Task coordination:* task coordination in MLOps involves managing various tasks within an ML workflow using Directed Acyclic Graphs (DAGs). These graphs establish the sequence of task execution based on their interdependencies, ensuring that each task is triggered at the correct stage.

*Data and model version control:* version control is crucial for tracking different versions of data, models, and code. This not only supports replicability but also aids in maintaining records for compliance and auditing purposes, providing traceability across project developments.

*Enhanced collaboration:* fostering a collaborative environment is essential for effective MLOps. This principle involves technical tools and practices that enhance teamwork across data, model, and code, reducing barriers between different functional areas and promoting a cooperative work culture.

*Ongoing model training and assessment:* continuous model training involves regularly updating the ML model with new data to maintain its relevance and accuracy. This principle includes automated re-training and evaluation processes that assess changes in model performance, ensuring that the model adapts to new data and contexts efficiently.

*Tracking and logging of ML metadata:* metadata tracking and logging are integral for recording detailed information about each training job within the ML workflows, such as the training parameters, performance metrics, and the data and code utilized. This comprehensive logging supports full traceability of all experimental runs.

*Continuous performance monitoring:* regular monitoring of data, models, code, and infrastructure is necessary to identify and correct any issues that might affect product quality. This involves tracking the operational performance of systems and models to ensure that they continue to meet expected standards.

*Implementing feedback loops:* establishing effective feedback loops is crucial to integrate insights gained from performance assessments into the development process. This includes adjustments based on model performance and iterative improvements from later stages of model development back to earlier stages.

*Containerization and microservices:* a container image is a small, standalone executable package that contains everything required to run an application, including the code, runtime, system tools, system libraries, and configurations (Alvaro Luis et al., 2023). Containers are more space efficient, allowing for the management of more applications with fewer virtual machines and operating systems (Alvaro Luis et al., 2023). Containers are very helpful in ML projects, given that they ensure consistency, improve resource efficiency, simplify deployment, and enhance scalability and flexibility. Containers wrap ML environments, letting the practitioners focus on developing and optimizing their models instead of managing complex dependencies and infrastructure. Containers facilitate the development of ML applications as microservices, allowing different components (e.g. data preprocessing, model training, dashboarding) to be developed, deployed, and scaled separately. Microservices are a method for developing a single application by breaking it down into a collection of small, independent services. Each service runs in its own process and communicates using lightweight protocols, often through an HTTP resource API. These services are designed around specific business functions and can be deployed independently using fully automated deployment tools. With minimal centralized management, microservices can be implemented in various programming languages and utilize different data storage solutions. They are independently scalable, allowing them to be updated and replaced without affecting the entire system, ultimately accelerating the release process (Pautasso et al., 2017).

*Industrial MLOps* represents the specialized adaptation of machine learning operations for industrial environments, integrating edge computing with AI/ML deployment pipelines to address sector-specific challenges. It is an approach that focuses on the automation and operationalization of AI development, including model packaging, monitoring, and deployment, where edge computing enables decisions at data sources rather than sending data to centralized databases (Rani et al., 2024). This approach addresses key industrial concerns including resource constraints on edge devices, data availability and quality issues, latency requirements, and security compliance. Comprehensive Industrial MLOps architecture includes data collection from industrial IoT sensors, preprocessing, model training, deployment using containerization technologies, continuous monitoring for model drift, and governance (Rani et al., 2024). Industrial MLOps enable various applications across the manufacturing, healthcare, automotive, agriculture, and smart city domains, providing organizations with improved decision-making capabilities, real-time optimization, and predictive maintenance, ultimately addressing the intrinsic technical debt of industrial ML systems (Chatterjee et al., 2022a; Garrone et al., 2023; Rani et al., 2024).

### **2.3. Scope and related work**

Recent MLOps literature has produced numerous survey papers and general overviews that examine the field from broad, domain-agnostic perspectives. Papers such as Anas et al. (2023), Andrew Tamburri (2020), Kreuzberger et al. (2023), Sasu Makinen et al. (2021), and Symeonidis et al. (2022) represent this trend, providing comprehensive reviews of MLOps definitions, tool landscapes, maturity models, and common implementation challenges.

Similarly, Testi et al. (2022) propose a taxonomy for clustering MLOps research and present a ten-step methodology covering business understanding through sustainability. Steidl et al. (2023) and Mboweni et al. (2022) further exemplify this systematic review approach, with the former developing a four-stage pipeline framework through analysis of 151 sources and practitioner interviews, while the latter conducts a systematic review of 60 studies to examine definitional variations and identify knowledge gaps in MLOps literature. Both studies emphasize the lack of standardized definitions and a common understanding across the field. As shown in Table 1, these studies focus primarily on general MLOps challenges without manufacturing-specific implementation. Diaz-de-Arcaya et al. (2024) extend this domain-agnostic trend through a joint systematic survey of both MLOps and AIops methodologies, analyzing 93 studies to

**Table 1.** MLOps literature comparison.

Study	Year	Methodology	Focus	Manufacturing specific	Implementation and validation	Challenges and solutions
Kreuzberger et al. (2023)	2023	Literature Re-view	Overview of MLOps principles	No	None	General MLOps challenges
Testi et al. (2022)	2022	Taxonomy Dev.	Categorization of MLOps pipeline types	No	None	Classification challenges
Diaz-de-Arcaya et al. (2024)	2023	SLR	MLOps and AIOps deployment strategies	No	Limited case studies	General deployment challenges
Mboweni et al. (2022)	2022	Systematic Review	Standardizing MLOps terminology	No	None	Definitional and standardization
Faubel et al. (2023)	2023	SLR + Project Exp.	Industry 4.0-specific MLOps challenges	Yes	Project-based insights	Industry 4.0-specific vs. general MLOps
Faubel and Schmid (2024)	2024	Multiple Case Study	Study MLOps Practices in three companies	Yes	Interview-based observation	Implementation challenges through external observation
Faubel et al. (2025)	2025	Multi-company Collab.	Industrial Partners collaboration on CPPS MLOps	Yes	Collaborative research documenting experiences	CPPS-specific solutions through partnerships
Gulshat A et al. (2024)	2024	Literature Review	ML algorithms applied to Digital Twin monitoring	Yes	Conceptual Framework with examples	Digital Twin-specific ML integration
Kolar Narayanappa and Amrit (2024)	2024	SLR + Interviews	Implementation challenges preventing MLOps adoption	No	Interview validation with 12 practitioners	Organizational, technical, and operational barriers
Steidl et al. (2023)	2023	MLR + Interviews	Framework development for AI pipeline management	No	Interview validation only	Pipeline implementation challenges
Chakraborty et al. (2025)	2025	Systematic Mapping	Comprehensive mapping of MLOps research trends	No	Analysis of 32 Studies across domains	Pipeline-specific challenges
Our study	2025	SLR + Thematic	Manufacturing or maintenance	Yes	Full implementation + real empirical data	Manufacturing specific

identify shared challenges in cross-functional collaboration, data management complexity, and infrastructure orchestration across diverse computational environments from cloud to edge. Their work emphasized that MLOps thrive in traditional industrial environments while AIOps flourish in challenging IT operations contexts like 5G/6G networks, yet both methodologies face similar fundamental challenges in operationalizing AI solutions. Taking a complementary empirical approach, Kolar Narayanappa and Amrit (2024) combine systematic literature review with grounded theory analysis of interviews with 12 ML practitioners to identify specific implementation barriers, organizing their findings into four dimensions: organizational challenges (human resources, user resistance, slow processes), technical challenges (infrastructure, standards), operational challenges (deployment complexities), and business challenges (value demonstration, budget constraints). Some recent work has begun exploring MLOps applications in specific domains, such as Gulshat A et al. (2024), who examined ML algorithm integration within digital twin monitoring systems, highlighting the importance of MLOps practices for ensuring reliable deployment and lifecycle management of ML models in cyber-physical systems. Chakraborty et al. (2025) conducted a systematic mapping study of 32 studies to identify challenges across three MLOps pipelines: data manipulation, model creation, and deployment. Their analysis reveals that model deployment pipeline challenges dominate the literature, particularly focusing on model monitoring, managing deployment pipelines, and operations feedback loops, while data management and model creation receive comparatively less attention.

These contributions offer valuable foundational understanding of MLOps principles and available toolsets, but they largely focus on enterprise software contexts without addressing the specific constraints and complexities that arise in manufacturing environments.

The comparison presented in Table 1 highlights a notable gap in existing literature. Faubel et al. have conducted valuable, extensive research on MLOps in Industry 4.0 contexts through systematic literature reviews (Faubel et al., 2023), case studies of three large companies (Faubel & Schmid, 2024), and collaborative documentation of cyber-physical production systems (Faubel et al., 2025). Their work identifies Industry 4.0-specific MLOps challenges and documents implementation approaches across

electronics, metal production, and chemical processing industries through observational research and industry partnerships. While these studies provide valuable theoretical insights and document existing practices across multiple industrial domains, there remains a need for a review of empirical studies that involve direct implementation of MLOps architecture with real production data. Such hands-on implementation studies could complement the existing observational research by providing practical validation of MLOps approaches in manufacturing environments.

Understanding ML systems is comparatively more difficult than traditional software architectures due to the complex interplay between software development, data science, and data engineering (Sculley et al., 2015). When applied to specific domains, such as manufacturing, the complexity increases further. Existing literature, particularly review articles, has extensively documented various aspects of MLOps, including its current state, automation frameworks, implementation tools and technologies (Woźniak et al., 2025), core components (Testi et al., 2022), implementation challenges and best practices (Bayram & Ahmed, 2025; Zarour et al., 2025), and approaches to ensure trustworthiness in MLOps systems (Bayram & Ahmed, 2025). A major portion of this literature is in the context of software engineering and AI systems deployment. This article is driven by the lack of comprehensive studies that examine the holistic development, deployment, and maintenance of ML and AI technologies, especially within the manufacturing sector and its associated challenges. Therefore, our study distinguishes itself from existing literature by focusing specifically on MLOps architectures in real-world industrial manufacturing and maintenance contexts. Rather than providing a general overview, we conduct an in-depth analysis of characteristic features of MLOps deployments in industrial settings, explore domain-specific adoption challenges, and map architectural features to practical recommendations for overcoming these challenges in manufacturing environments. As evidenced in Table 1, our approach fills a critical gap by studying the work that focuses on full implementation with real empirical data in manufacturing-specific contexts.

The scope of this research was carefully shaped through thoughtful inclusion criteria designed to ensure both practical relevance and empirical rigor. Rather than casting a wide net, this review embraces a quality-focused approach with domain-specific selection criteria that looked for studies focusing on the following:

- Clear manufacturing or maintenance context with specific industry applications.
- Detailed MLOps architectural proposals that provide concrete technical insights.
- Evidence of real-world deployment or clear readiness for implementation, with deployment as the logical next step.
- Focus on manufacturing-MLOps challenges that specifically emerge when operationalizing ML models in industrial settings, rather than addressing generic MLOps or manufacturing challenges separately.

This implementation-focused approach naturally led to excluding theoretical papers, generic enterprise MLOps frameworks, and manufacturing digitization studies that did not address the unique convergence challenges of implementing MLOps architectures in manufacturing environments. The selection process prioritized empirically grounded architectures that either documented real-world manufacturing deployment experiences or were specifically developed to address domain-specific implementation challenges such as OT/IT integration complexities, edge-cloud hybrid deployment in industrial settings, and manufacturing-grade model operationalization requirements. This distinctive approach, as outlined in Table 1, sets our study apart from previous works that either lack manufacturing specificity or provide limited implementation validation. This approach is consistent with evidence-based practice principles in applied research, where focused samples of highly relevant, cross-domain implementation-ready studies tend to offer more actionable insights than broader samples that include preliminary work addressing MLOps and manufacturing challenges in isolation.

### 3. Methodology

#### 3.1. Systematic literature review protocol

The systematic literature review (SLR) was conducted following an adapted version of the preferred reporting items for systematic reviews and meta-analyzes (PRISMA) guidelines (Moher et al., 2009).

Unlike existing MLOps surveys that focus on general enterprise contexts, this methodology was specifically designed to address the critical gap in manufacturing-specific MLOps implementations. The approach distinguishes itself by examining the unique convergence challenges that emerge when operationalizing ML models in industrial settings, rather than addressing generic MLOps or manufacturing digitization challenges separately.

The protocol was developed to prioritize quality over breadth, using implementation-focused selection criteria that target empirically grounded architectures with real manufacturing deployment evidence or clear implementation readiness. This quality-focused approach with domain-specific selection ensures practical relevance and empirical rigor, distinguishing it from theoretical MLOps frameworks or observational manufacturing studies.

### 3.1.1. Information sources and manufacturing-specific search strategy

Scopus was selected as the primary database for this systematic literature review due to its comprehensive indexing of publications from major academic publishers and societies, including IEEE, ACM, Elsevier, Springer, Taylor & Francis, and others. This provides extensive interdisciplinary coverage of engineering, computer science, and manufacturing literature essential for capturing MLOps implementations that bridge these domains (Chou, 2012). The search strategy specifically targeted manufacturing-MLOps convergence rather than generic MLOps applications. After iterative refinement to balance sensitivity and specificity, the final search string captured the intersection of MLOps terminology with manufacturing contexts:

*TITLE-ABS-KEY (('ML OPS' OR 'MLOps' OR 'ML-OPS' OR 'machine learning operations') AND ('factory' OR 'manufac\*' OR 'maintenance' OR 'industr\*'))*.

The applied filters included subject areas (Engineering, Computer Science, Mathematics, Decision Sciences), English language restriction, and the search was conducted on March 20, 2025, yielding 186 initial records.

### 3.1.2. Implementation-focused eligibility criteria

Systematic screening was documented in a software called Rayyan (Ouzzani et al., 2016), where the documents were labeled according to the reason for inclusion, exclusion, and comments. This study employed stringent selection criteria designed to identify manufacturing-MLOps convergence studies rather than generic enterprise applications.

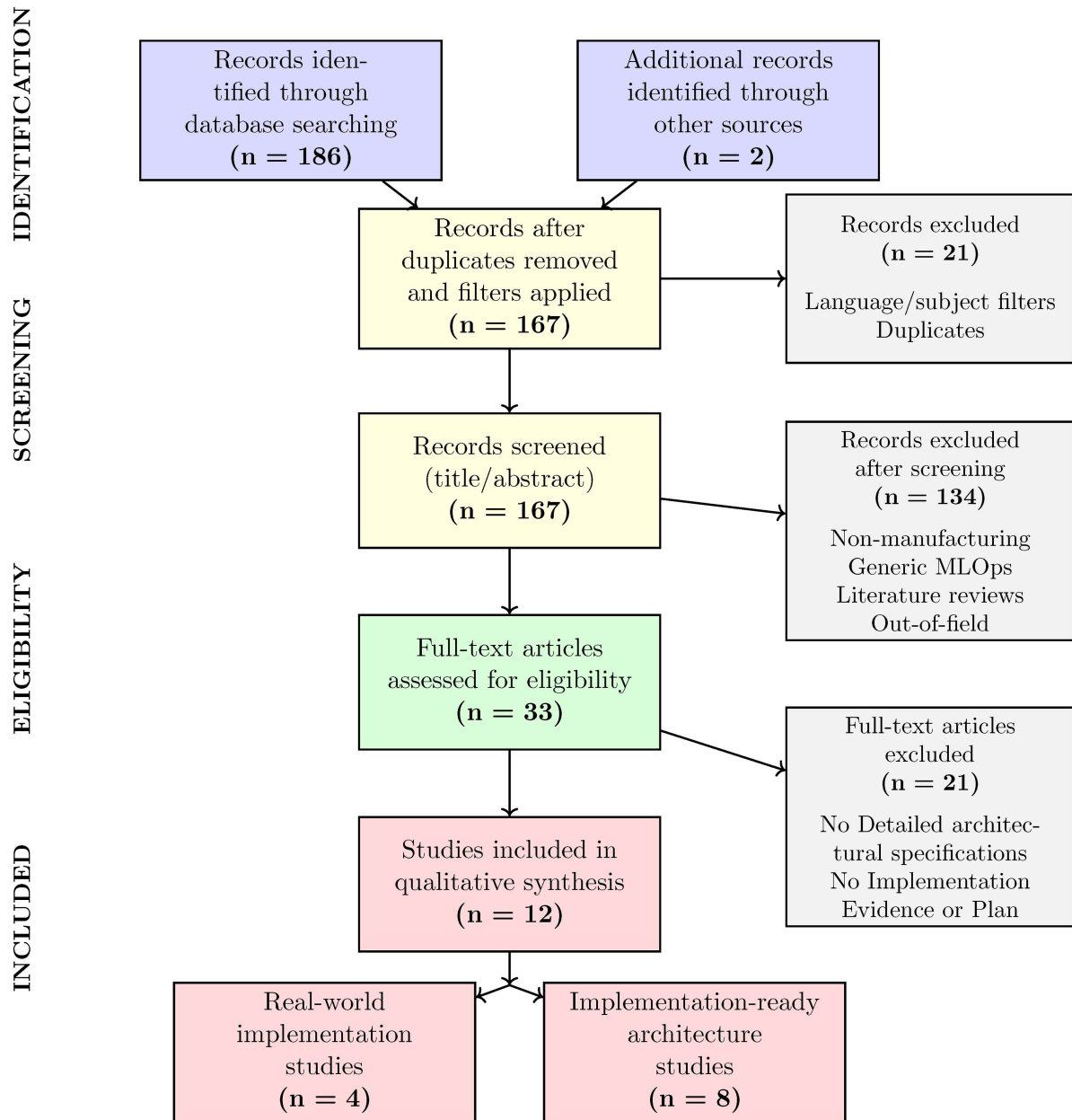
Inclusion criteria:

- Studies labeled as 'applied MLOps to Industry use case' with manufacturing or maintenance contexts.
- Articles proposing detailed MLOps architectural frameworks with concrete industrial implementations.
- Research demonstrating real-world deployment evidence or implementation-ready architectures.
- Studies addressing manufacturing-specific MLOps challenges and architectural solutions
- Papers with an explicit focus on industrial MLOps convergence rather than generic enterprise contexts.

All studies not meeting the above inclusion criteria were excluded. Additionally, the following specific exclusion criteria were applied:

Exclusion criteria (with examples from screening):

- *Generic enterprise MLOps frameworks*: studies labeled 'no manufacturing or maintenance related stuff' or 'general concepts and frameworks about MLOps.'
- *Domain misalignment*: papers focused on healthcare, finance, telecom, insurance, gaming, or other non-manufacturing sectors.
- *Literature reviews*: studies marked as 'literature review' without original architectural contributions.
- *Insufficient MLOps focus*: papers labeled 'MLOps as a future step but not realized' or lacking MLOps implementation details.
- *Technical debt focus*: studies primarily addressing technical debt management rather than operational deployment.



**Figure 1.** PRISMA flow diagram showing the systematic selection process for industrial MLOps literature review.

- *Publication type exclusions:* conference abstracts, book chapters, workshop reports without primary research data.

The selection process prioritized empirically grounded architectures through systematic screening documented in Rayyan, as illustrated in Figure 1:

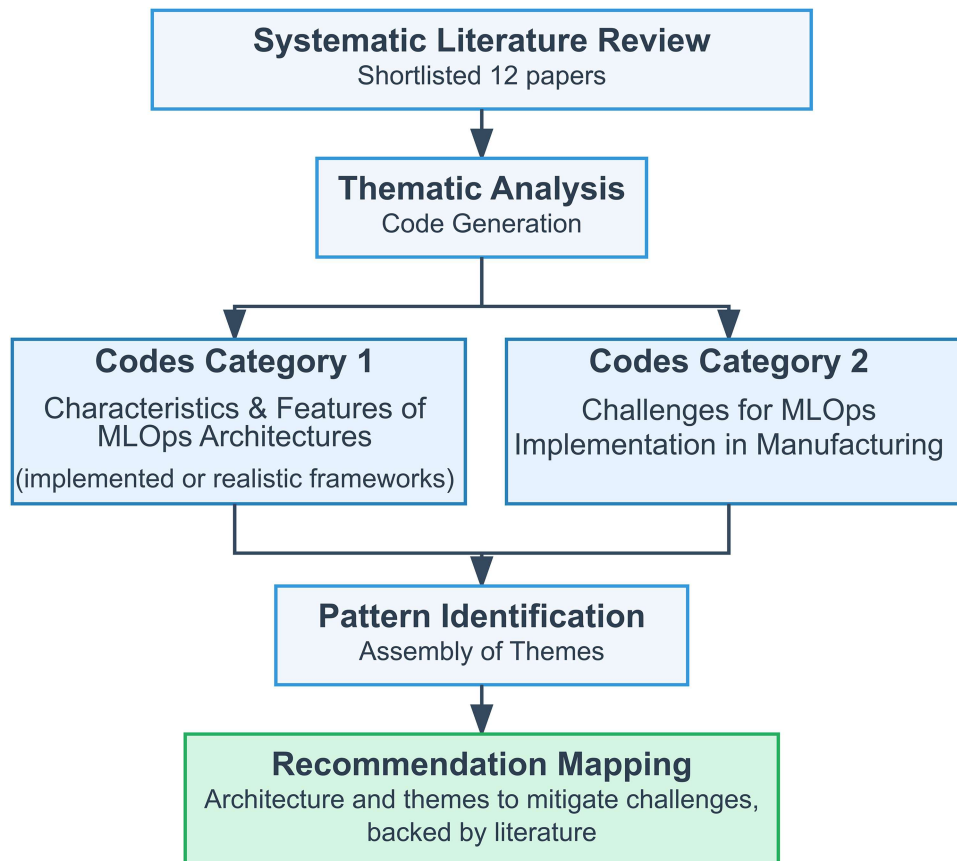
### 3.1.3. Three-stage implementation-focused selection

- Initial Screening: Title/abstract screening identified manufacturing-MLOps convergence studies (167 records after filters, 134 excluded for reasons including ‘no manufacturing or maintenance related stuff,’ ‘general concepts about MLOps,’ and domain misalignment)
- Full-text Assessment: Detailed evaluation focused on architectural depth and implementation evidence using predefined labels including ‘applied MLOps to Industry use case,’ ‘Manufacturing-MLOps challenges,’ and ‘Architecture and framework to MLOps’ (33 assessed, 21 excluded)

- Quality Assessment: Final selection based on real-world deployment validation or implementation readiness, resulting in 12 studies (10 selected plus 2 via backward snowballing (Wohlin, 2014))

### 3.2 Thematic analysis of architectures and challenges

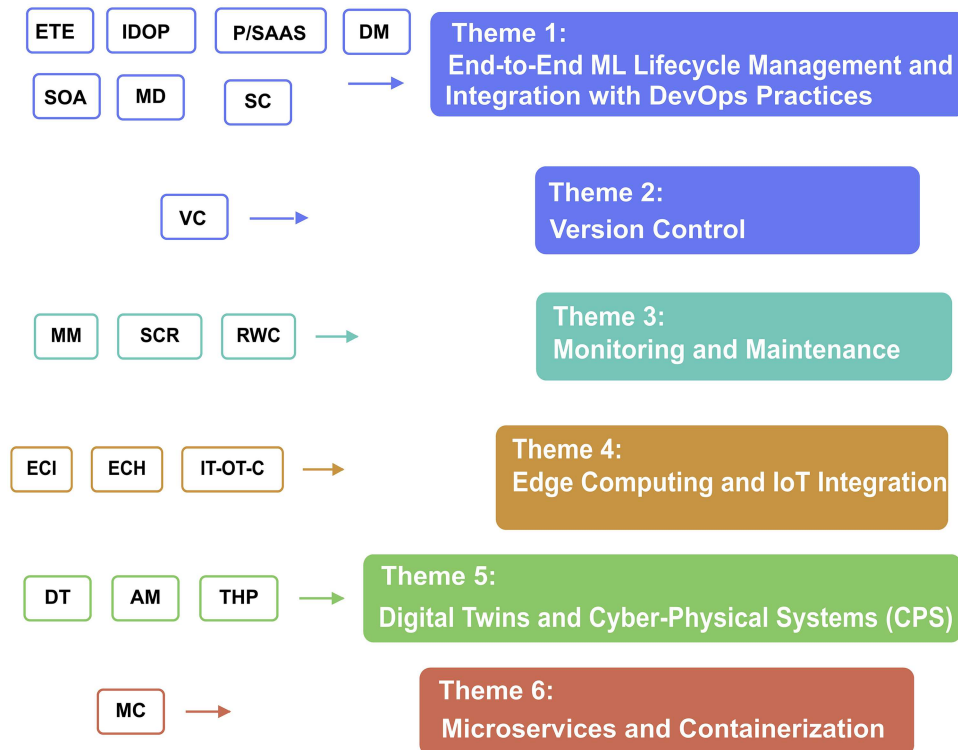
To systematically analyze and synthesize information from the selected articles, a thematic analysis was performed following the structured approach outlined by Braun and Clarke (2006) (Scharp & Sanders, 2019). The methodology followed the structure highlighted in Figure 2. This analysis identified key themes related to MLOps architectures implemented in real-world industrial settings, as well as those planned for specific manufacturing environments or presented as realistic conceptual frameworks. Additionally, these themes highlighted various challenges emphasized by the authors across these implementations. The process began with thorough familiarization with the data, where each article was carefully read and reread to gain a more in-depth understanding of its content. Key details related to architectures, methodologies, and challenges were highlighted and documented. Notes were taken to capture initial impressions and significant insights. The 12 selected articles were carefully reviewed to extract relevant details about the architectures proposed to operationalize machine learning in industrial settings and the challenges associated with ML implementation and these architectures. A software called Rayyan (Ouzzani et al., 2016) was used to better scan, handle, and document notes regarding initial impressions and extracted data. Following familiarization, the next step involved generating initial codes. Thematic coding was applied to categorize the extracted data. Codes for architectures were created to represent recurring design principles, methodologies, and system components (Code Category 1, Table 2) and codes for challenges were developed to capture specific issues and barriers (Code Category 2, Table 4) in implementing MLOps in industrial contexts and with regard to the components of the architectures. Each code was systematically applied across the articles to ensure consistency and comprehensiveness. After initial coding, the process



**Figure 2.** An overview of the methodology followed in this study.

**Table 2.** Category 1 – feature code abbreviations.

Abbreviations	Codes
AM	Study incorporates additive manufacturing processes integrated with MLOps workflows
DM	Study emphasizes data management strategies, including data storage, processing, and governance within the proposed architecture
DT	Study integrates digital twin technology with MLOps for virtual representation and simulation of physical systems
ECH	Architecture employs an edge-cloud hybrid deployment model, distributing computational workloads between edge devices and cloud infrastructure
ECI	Study integrates edge computing capabilities with Internet of Things (IoT) devices for distributed ML inference
ETE	Architecture supports end-to-end machine learning lifecycle management, from data collection through model deployment and monitoring
IDOP	Study integrates DevOps principles and practices into the MLOps workflow for continuous integration and deployment
IT-OT-C	Architecture bridges information technology (IT) and operational technology (OT) systems, enabling convergence of digital and physical operations
MC	Architecture utilizes microservices-based design and containerization technologies (e.g. Docker, Kubernetes) for modular deployment
MD	Study employs modular design principles, allowing independent development and deployment of architectural components
MM	Architecture includes monitoring and maintenance capabilities for tracking model performance and system health
P/SAAS	Solution is delivered as a platform-as-a-service (PaaS) or software- as-a-service (SaaS) offering
RWC	Study already demonstrates implementation or validation through real-world industrial use cases (not the next step, but already implemented)
SC	Architecture designed with scalability to handle increasing data volumes and computational demands
SCR	Study applies shape-constrained regression to enforce domain- specific constraints in model predictions
SOA	Architecture follows service-oriented architecture (SOA) principles with loosely coupled, reusable services
THP	Study utilizes the Thingier. An IO platform as infrastructure for implementing MLOps capabilities
VC	Architecture incorporates version control systems for tracking changes to code, models, and data

**Figure 3.** Assembly of architecture feature themes from codes.

moved to searching for themes, where patterns were identified from the initial codes and assembled into themes (Figures 3 and 4). Architectural themes captured patterns like scalability, edge deployments, and integration of digital twins, while challenge themes reflected recurring issues such as data management difficulties, heterogeneity in manufacturing environments, and non-technical barriers like cost and skills shortages. The identified themes were then reviewed to ensure they accurately represented the data and aligned with the research questions. A comprehensive coding framework with specific codes and their meanings guided the thematic analysis throughout the process. The verification process was strengthened



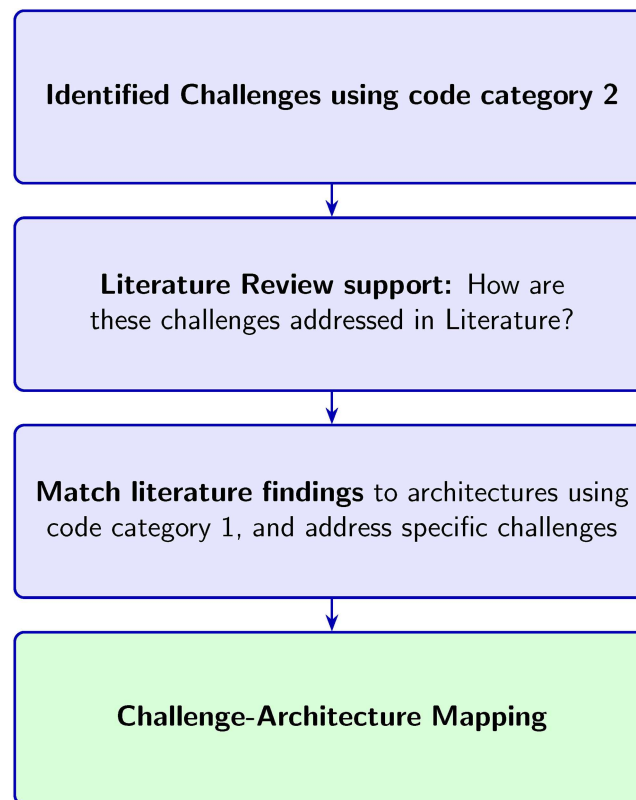
**Figure 4.** Assembly of challenge themes from codes.

through several established validation methods. A detailed audit trail was kept documenting the decision-making process during coding and theme development. The themes were validated through regular consultation sessions with research supervisors, which provided external verification of the analytical process. The analysis was further validated by comparing the themes against the complete dataset to verify comprehensive coverage. These verification procedures ensured the reliability of the thematic analysis, which enhanced the systematic review by revealing deeper patterns and connections in the literature.

The recommendation mapping process methodically connected suitable MLOps architectural themes to identified manufacturing challenges following the methodology highlighted in Figure 2. The process was literature-driven and evidence-based (Figure 5):

First, a comprehensive review of existing literature beyond the 12 selected articles was conducted to understand how researchers and practitioners have addressed similar MLOps challenges in various contexts. This broader literature review provided insights into proven approaches and established best practices for tackling each identified challenge.

Second, using the architectural feature codes from Category 1, identified through thematic analysis, we systematically examined which of the 12 reviewed architectures possessed characteristics aligned with the literature-supported solutions.



**Figure 5.** Recommendation mapping methodology.

Not all architectures addressed all challenges; only those with relevant features and explicit evidence were included in the mapping. This selective, evidence-based approach ensured that recommendations were grounded in both the broader MLOps literature and the specific industrial implementations examined in this review. The resulting mapping (visualized in [Figure 6](#)) creates direct connections between implementation barriers and architectural solutions, providing practitioners with literature-validated pathways for MLOps adoption in manufacturing settings.

## 4. Results

### 4.1. Findings regarding MLOps architecture

As discussed in [Section 3](#), all the articles chosen for this literature review are about industrial applications, especially in manufacturing or maintenance. MLOps architectures in four articles were applied and tested in a real-world industrial setting (Alvaro Luis et al., 2023; Antonini et al., 2022; Bachinger et al., 2024; Venanzi et al., 2023). Six articles do not explicitly mention that the MLOps architecture was tested and implemented in a real-world use case (Chatterjee et al., 2022a, 2022b; Hegedus & Varga, 2023; Cha et al., 2023; Safdar et al., 2024; van Bruggen et al., 2024). However, a detailed description of the development and design of an MLOps platform is tailored and targeted to a specific manufacturing environment, and it is suggested that the next step is to evaluate the ability of the architecture to be implemented. While Raffin et al. (2022) provide a realistic conceptual framework and deployment view, MLOps architecture discussed by Martel et al. (2020) is a conceptual framework derived from the analysis of various real-world use cases and best practices. Each article shows different use cases within this field, so the respective MLOps architecture discussed by the mentioned authors is customized to fit the specific needs of deploying and managing ML models in each unique industrial setting. Although there is uniqueness in each application, common themes can be observed in architecture. [Table 3](#) represents the feature comparison across

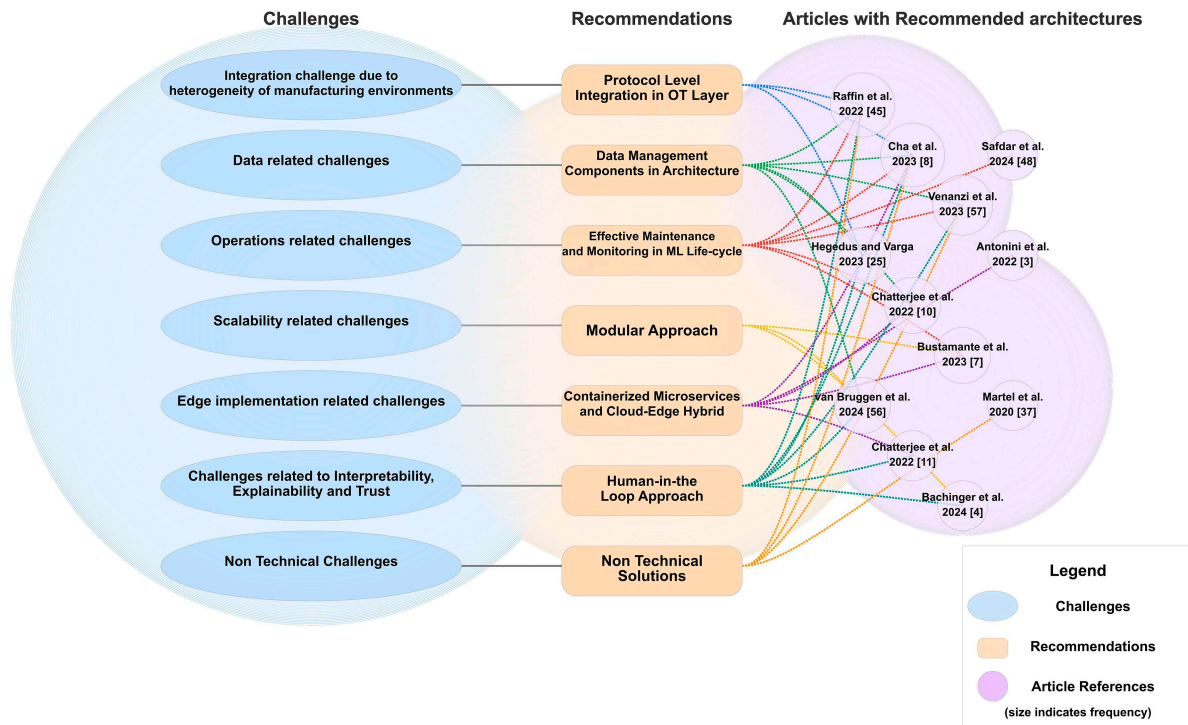


Figure 6. Architecture recommendation network map.

Table 3. Feature comparison across architectures proposed/discussed by different authors.

Authors, year	AM	DM	DT	ECH	ECI	ETE	IDOP	IT-OT-C	MC	MD	MM	PAAS	RWC	SAAS	SC	SCR	SOA	THP	VC
Raffin et al. (2022)		✓		✓	✓	✓	✓	✓	✓	✓	✓								✓
van Bruggen et al. (2024)			✓			✓	✓	✓		✓	✓				✓	✓	✓		✓
Bachinger et al. (2024)		✓											✓						
Cha et al. (2023)		✓				✓	✓		✓	✓	✓								✓
Venanzi et al. (2023)		✓			✓	✓	✓	✓	✓	✓	✓		✓						✓
Safdar et al. (2024)	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓				✓				
Chatterjee et al. (2022a)				✓	✓	✓	✓	✓	✓	✓	✓								✓
Alvaro Luis et al. (2023)				✓	✓	✓	✓	✓	✓	✓	✓		✓	✓				✓	
Martel et al. (2020)					✓	✓	✓	✓	✓	✓	✓		✓	✓					
Hegedus and Varga (2023)			✓			✓	✓	✓			✓				✓				
Chatterjee et al. (2022b)			✓			✓	✓	✓			✓								
Antonini et al. (2022)					✓	✓	✓	✓			✓					✓			

architectures proposed/discussed by different authors. Figures 3 and 4 represent the evolution of the initial codes into themes and pattern identification.

*End-to-End ML lifecycle management and integration with DevOps practices:* Integration with DevOps practices, along with the establishment of end-to-end ML lifecycle management, is an important aspect in modern MLOps architectures in industrial applications. End-to-end ML lifecycle management is a fundamental theme that ensures the entire process, from data collection and preprocessing to model deployment and monitoring, is managed smoothly. Integration with DevOps practices involves adopting the core principles of DevOps, like continuous integration, continuous delivery, and, in some cases, continuous training and version control to streamline the machine learning lifecycle. It is evident from the findings that these two aspects could be seen to some degree in all the architecture from all the articles, thus pointing out to be the backbone and basic necessity for MLOps architectures in industrial settings. Some architectures depict high levels of automation in all the stages of the lifecycle. For example, the architecture discussed by Bachinger et al. (2024) highlights the importance of automating ML training pipelines to reduce human intervention. The automation here is incorporated into all the stages, including data preprocessing, feature extraction, model training, and validation, enabling scalable and efficient model development. Raffin et al. (2022) stress on ensuring each stage of the ML lifecycle is meticulously managed,

from initial data ingestion to final model deployment and continuous monitoring. Additionally, Safdar et al. (2024) highlight the need for a consistent lifecycle management system that integrates data from various sources and stages of the manufacturing process. The platform addresses the complexities of managing the ML lifecycle in a specific type of industrial setting.

*Version control:* Version control is an important component of MLOps architecture to manage the lifecycle of machine learning models, data, and related artifacts. The architecture discussed in all the reviewed articles demonstrates how version control is employed in their respective cases. Additionally, the MLOps architectures discussed in the selected articles have employed version control systems for various purposes like tracking changes, smooth collaboration, reversibility, accountability, security/backup, and scalability. As mentioned above, version control is the core principle derived from DevOps practices. It could be understood from the findings that some architectures explicitly showcase version control methods (Alvaro Luis et al., 2023; Chatterjee et al., 2022a; Cha et al., 2023; Raffin et al., 2022; van Bruggen et al., 2024; Venanzi et al., 2023).

Raffin et al. (2022) discussed the architecture that employs an artifact repository to store versioned models and configurations to ensure traceability and reproducibility. The above-mentioned repository enables the tracking of different versions and helps in the retrieval of a particular iteration if needed. This versioning process is automated using a continuous integration pipeline where trained models and related software artifacts are containerized, tested, and committed to a version control system, as van Bruggen et al. (2024) in their architecture uses version control as a central entity for managing machine learning workflows within the digital twin and MLOps platform. The architecture employs the Model Package Group Store (MPGS) as a model registry to store various versions of trained models along with their metadata, and the Pipeline Package Group Store (PPGS) as an ML pipeline registry to store various versions of ML pipelines. This ensures that each digital twin can manage its own version-controlled workflows independently.

Cha et al. (2023) describe the utilization of Git for version control. The platform ensures that any modifications in the codebase or deployment configurations are systematically tracked. When changes are made to the code, developers submit these updates to a central place called GitHub. This action, known as a push or pull request, automatically starts a process in a tool called Jenkins. Jenkins takes the updated code and packages it into a single file. This packaged code is then uploaded to a storage service called Amazon S3, which is like a cloud-based hard drive. Once the code is safely stored in Amazon S3, another service called AWS CodeDeploy steps in. It fetches the packaged code from Amazon S3 and installs it onto servers running the application, known as AWS EC2 servers. This entire process ensures that updates to the application can be deployed quickly and consistently across different environments. If something goes wrong, it is easy to revert to a previous version, much like undoing changes in a document, which helps keep the application stable and reliable.

Venanzi et al. (2023) in their architecture talk about the Bi-Rex platform that employs a component called the MLOps-DevOps Controller (MDC) to facilitate version control and automate the deployment of updated services and models. This controller includes an update controller (UC) module responsible for monitoring a repository for new versions of machine learning models or services. When an update is detected, the UC downloads the new version and triggers the MIINT middleware module to deploy the updated components on the DEEMON edge nodes. This process ensures that the edge devices always run the latest and most accurate models and services, thus taking care of issues like model drift and model performance degradation.

Chatterjee et al. (2022a) described the steps taken for efficient version control adopted in its architecture called the 'IoT-edge/fog-cloud architecture.' To begin with, ML models are trained and validated using historical data, often employing tools like Python scripts or Jupyter notebooks. Later, the validated model, along with its dependencies, is encapsulated into a Docker container, which is defined by a Dockerfile. Next is the building of this container, tagging it with the version number and pushing it into a Docker registry (Docker Hub). Kubernetes is used for orchestration, which pulls the Docker image from the registry and deploys it to the specified environment. The system is tasked with continuously monitoring the performance of the model using tools like Prometheus. If there is any degradation of the model detected, the architecture triggers a retraining through CI/CD pipelines such as Jenkins or GitLab CI/CD. The updated model goes through the whole process, which is stated above, using a rolling update strategy,

allowing a smooth version control and equipping the architecture to revert to previous versions if needed. Bustamante et al. (2023) adopted a very similar approach in their architecture but emphasized the use of Thinger.io's integrated platform for version control.

*Monitoring and maintenance:* all reviewed articles adopt various steps for monitoring and maintenance across different stages of the MLOps lifecycle. These include initial incoming data monitoring, service logs tracking, pipeline monitoring, infrastructure monitoring, manufacturing process monitoring, model performance monitoring, etc.

Raffin et al. (2022) described 'monitoring services' and 'central feature store' in their proposed MLOps architecture to keep track of degradation or changes in the accuracy of the model over time and detect shifts in data patterns that may indicate drift. They also mention the importance of employing 'drift detection algorithms' that can identify changes in data distribution or model performance metrics. Analytics tools are used to visualize model performance and detect anomalies in real time. This architecture supports automated retraining pipelines that are triggered when significant drifts are detected. The architecture manages the deployment and scaling of models and services using orchestration frameworks such as Kubernetes which makes the retraining process easier by automating the deployment of updated models. Model management platforms like MLflow are used to oversee the entire lifecycle of a model, from training to deployment and monitoring, ensuring everything runs smoothly.

Cha et al. (2023) describe a MLOps architecture that leverages technologies such as Kube-Flow and Kubernetes to enable retraining and adaptation of machine learning models. These platforms enable automated retraining pipelines that can be triggered by detected anomalies or changes in data patterns. The architecture also offers a framework for model validation and feedback. To validate models, the system integrates data from various sources, including RGB and ToF cameras. Convolutional neural networks (CNNs) are used to classify products as normal or defective, ensuring high accuracy and reliability in models.

Venanzi et al. (2023) describe the Bi-Rex platform that supports the automatic deployment of machine learning models and services using the MLOps-DevOps controller (MDC), reducing the need for human intervention and ensuring that models are always up-to-date. The Bi-Rex platform uses the Siemens Industrial Edge to manage data collection and processing on the shop floor. This integration helps with easy monitoring and maintenance of models deployed in various industrial environments. The platform continuously checks model drift and performance degradation. If the system detects a drift or a performance drop, it automatically retrains the model with new data and updates the deployed model. Although the platform automates many processes, it also allows for supervision and intervention as needed. Operators can initiate model updates or service deployments manually, providing a balance of automation and human control.

Chatterjee et al. (2022a) quote ways to implement QA in MLOps within industrial environments. They talk about modular QA architecture that includes defining QA dimensions, selecting relevant dimensions, and iteratively training QA models. This modular approach allows for real-time monitoring, anomaly detection, drift adaptation, and real-time QA assessments, allowing for continuous evaluation of data quality and model performance to make sure that machine learning models remain reliable. The QA assessments include drift detection as one of the dimensions which is an important monitoring and maintenance aspect. The framework also supports automated retraining and deployment of models using containerization tools like Docker and Kubernetes, enabling integration of new data and model updates, thus enabling a solid model maintenance and monitoring system. Bustamante et al. (2023) describe an interesting theme of framework that leverages Edge-cloud collaboration for efficient model maintenance and monitoring. The cloud deals with resource-intensive tasks such as training, and edge devices focus on real-time inference and monitoring.

In summary, the MLOps architectures and platforms referred to in the selected articles adopt key concepts like real-time monitoring, anomaly detection, feedback loops, automated retraining, and edge deployment for realizing functional model maintenance and monitoring which is the core requirement for successful implementation of MLOps architecture in an industrial setup.

*Edge computing and IoT integration:* the MLOps architecture described in six out of twelve reviewed articles incorporates edge devices within an IoT setup, highlighting their suitability for deploying ML models in an industrial environment (Alvaro Luis et al., 2023; Antonini et al., 2022; Martel et al., 2020;

Raffin et al., 2022; Safdar et al., 2024; Venanzi et al., 2023). Antonini et al. (2022) introduce a MLOps framework specifically targeting resource-constrained IoT devices. The framework consists of all phases of the ML life cycle, including local data preprocessing, model training, deployment, and continuous monitoring directly on the IoT devices. This approach makes sure that real-time analytics can be performed close to the data source, which is important for applications like anomaly detection in industrial machinery. Bustamante et al. (2023) explain in detail about a platform that facilitates the deployment of ML models from cloud environments to edge devices. Thinger.io provides a solid infrastructure for managing data streams, executing ML models, and monitoring performance metrics on edge devices. The referred platform supports essential features such as time-series data storage, file management, metadata handling, and event-driven processing, which are important for maintaining the reliability of edge-based ML operations. Venanzi et al. (2023) present an architecture that integrates edge computing capabilities with a big data platform to support adaptive analytics. The Bi-Rex platform is developed to handle the complexities of industrial IoT environments, providing a smooth workflow for data acquisition, preprocessing, model training, and deployment at the edge. The platform's architecture includes components like edge gateways and sensors that collect and preprocess data locally before transmitting it to the central system for further analysis. DEEPMON (Dynamic Edge computing for Plant Monitoring) is a component of the Bi-Rex Big Data platform designed to address typical issues found at the industrial OT layer. DEEPMON utilizes a microservices architecture, which means it is composed of small, independent services that each handle specific tasks (data processing and transformation, communication protocols, storage). DEEPMON plays an important role in enabling MLOps operations on edge nodes. It allows for the automatic deployment, monitoring, and updating of services and machine learning models, ensuring that edge devices can continuously adapt to changing conditions and requirements. This setup reduces latency and improves the responsiveness of the ML models, making it suitable for applications that require immediate insights, such as predictive maintenance and real-time quality control in manufacturing. The architecture referred to by Raffin et al. (2022) combines cloud and edge resources into a compact system, leveraging microservices, containerization, and orchestration for flexibility and scalability. Edge device interfaces between OT and IT, acquiring and preprocessing data, providing local inference for ML models, and logging activities. Cloud services perform time-insensitive tasks like monitoring, dashboarding, and incremental learning in this architecture. Cha et al. (2023) describe an approach to integrating edge computing and IoT into MLOps. This MLOps platform uses power consumption analysis, such as electricity and pneumatic energy, to optimize manufacturing processes in real time. Additionally, reviewed architectures explicitly show that hybrid architectures combining edge and cloud computing prove essential for effective MLOps implementation (Alvaro Luis et al., 2023; Chatterjee et al., 2022a, 2022b; Raffin et al., 2022; Safdar et al., 2024). Edge computing manages time-critical manufacturing tasks, while cloud infrastructure handles resource-intensive operations. This integrated approach optimizes resource utilization by strategically assigning tasks based on their time sensitivity, creating an efficient framework for MLOps adoption in industrial settings. Using these integrated MLOps systems helps industries to operate faster, more flexibly, and efficiently. This supports innovation and high performance in smart manufacturing and industrial IoT.

*Digital twins and cyber-physical systems (CPS)*: three articles explicitly discuss the usage and incorporation of digital twins in MLOps architecture (Hegedus & Varga, 2023; Safdar et al., 2024; van Bruggen et al., 2024). One interesting architecture that integrates digital twins with MLOps is presented by van Bruggen et al. (2024). This architecture shows the integration of digital twins with MLOps, which enables real-time prediction, optimization, monitoring, and control of physical assets. By incorporating AI into digital twins, these systems can independently analyze and learn from their environment, leading to better-informed and quality decision-making capabilities and operational efficiency. The architecture also points out the importance of managing digital twin interfaces (DTIs) with a twin digital twin architecture (TDTA). This architecture ensures that the digital twins are coupled with their MLOps modules, enabling easy and smooth scalability and management of ML workflows.

Adding on, another example of MLOps architecture involving digital twins is found in the context of Industry 5.0, as discussed by Hegedus and Varga (2023). This article leverages the integration of digital twins and CPS within an MLOps framework to facilitate real-time synchronization and efficient operations. The synchronization between digital twins and their corresponding physical modules is very

important for accurate predictions and effective model training. This integration makes sure that changes in the status parameters of one entity affect the other, which in turn is responsible for robust and transparent AI systems.

The architecture proposed by Safdar et al. (2024) discusses a cloud-based digital twin-enabled data management framework for additive manufacturing. The digital twin facilitates data exchange between the AM system and the MLOps platform to get accurate and up-to-date data. The digital twin enables virtual simulations of the AM process and the machine learning models in the MLOps framework can leverage the data from these simulations to improve predictive accuracy and optimize the AM processes for better performance. The feedback loop is essential in an MLOps framework for training the ML model which remains relevant over time. The digital twin gives a feedback mechanism by comparing actual results with the predicted outcomes, thereby allowing iterative improvements in machine learning models. This architecture also identifies the requirement of usage of edge computing and cloud computing resources for the deployment of the MLOps platform and for scalable data processing and storage, respectively.

*Microservices and containerization:* seven out of twelve reviewed architectures incorporate the concept of containerization and microservices as part of their proposed architecture for implementing MLOps in manufacturing settings (Alvaro Luis et al., 2023; Chatterjee et al., 2022a, 2022b; Cha et al., 2023; Martel et al., 2020; Raffin et al., 2022; Venanzi et al., 2023). Raffin et al. (2022) talk about how containerized microservice architecture helps in dealing with the heterogeneity of manufacturing systems, including varied communication protocols, diverse manufacturing processes, and different types of data. This approach paves the way for the encapsulation of functionalities into different services that can be independently developed, deployed, and scaled. This article recommends Kubernetes as a container orchestration tool to manage the deployment and scaling of containerized services. The architecture mentioned in this article makes use of containerized microservices to integrate edge devices and cloud instances. This integration makes it possible for time-sensitive operations to be handled at the edge and computationally intensive tasks to be processed in the cloud. The article clearly explains the nature of its architecture, being ‘event driven’ to enable communication and collaboration between microservices, particularly at the edge level. MQTT is mentioned as a lightweight message broker used for central event messaging, facilitating communication between services, data acquisition services (DAQS) in the architecture. The article refers to the use of container registries for storing containerized software artifacts.

Venanzi et al. (2023) primarily focus on the MLOps-DEVOps Workflow which uses DEEP-MON architecture at edge device and leverages microservices to perform specific tasks at the edge level with a modular approach where different microservices handle data enrichment, storage, visualization, and communication. DEEPMON runs microservices as containerized applications on edge devices. This includes using Docker to manage applications, allowing for easy deployment and updates across devices. The IT layer in the workflow uses a containerized model repository that stores the model images. The platform detects any model drifts/data drifts and trains a new model accordingly and stores it in the repository. This training could also be triggered manually as well. The platform further detects a new service/model in the repository and automatically deploys the containerized working models on the Siemens Industrial Edge device (IED).

Chatterjee et al. (2022b) describe an architecture that combines IoT, edge, and cloud computing for continuous delivery of machine learning software. The architecture uses microservices on edge nodes to support both real-time and offline data processing. These microservices handle tasks such as data collection, quality assurance, and action execution. Microservices deployment includes the use of containerization tools such as Docker and Kubernetes. The article discusses ‘action services,’ which are deployed as microservices. These services take specific actions based on the outcomes of ML model predictions and QA assessments. The article proposes a modular QA solution that uses microservices to manage various QA dimensions and ensure that each step in the QA process is handled correctly.

The architecture discussed by Bustamante et al. (2023) leverages containerization to package and deploy machine learning pipelines. There are two pipelines configured on ML workspace in Thinger.io platform, namely ‘ML training pipeline’ and ‘ML deployment pipeline.’ In the steps/nodes of the ‘ML training pipeline,’ preprocessing, training, and evaluation are encapsulated within a container, allowing for consistent execution across multiple environments. In the ‘ML training pipeline,’ ML models are packaged as container images, which are then pushed to a Docker registry and deployed on edge devices.

Containerization is the core part of this article's solution, whereas microservices with REST API are just quoted as a possible way to be used for model deployment.

Martel et al. (2020) propose a reference architecture for MLOps that includes container frameworks and microservices along with API framework as foundational elements and building blocks that relate to the infrastructure and cross-functional aspects.

Chatterjee et al. (2022b) propose an interesting architecture that combines edge and cloud services. They describe how to implement real-time decision-making capabilities at the edge node with containerized microservices. Microservices are mentioned as 'delivery and collection' and operate at the edge, receiving real-time data from pressure sensors attached to the electroslag remelting (ESR) process. This data is then transmitted to a microservice responsible for predicting vacuum pump pressure using a trained machine learning model. These edge microservices perform real-time analysis and prediction, whereas, on the other hand, cloud services handle tasks such as ML software training, retraining, and persistent data storage. When a new ML model is validated and meets the proposed testing criteria, it is automatically deployed to the edge to make real-time predictions. Docker is used for containerizing applications, and Apache Kafka is mentioned as an example of a tool for developing real-time data streaming pipelines. The article also discusses the future possibility of developing microservices for detecting model degradation and automating retraining processes.

#### 4.2. Challenges in integrating MLOps methodology

The evolving area of MLOps is tested by various challenges across different industrial and technological contexts. The authors of each reviewed article highlight specific challenges or gaps in machine learning or automated ML, serving as the motivation for proposing architectures related to automated ML or MLOps. These challenges can be broadly categorized as below with the help of code findings in Table 4. Table 5 represents the challenges discussed by various authors.

*Challenge 1 – integration challenge due to heterogeneity of manufacturing environments:* manufacturing systems are highly diverse, utilizing various technologies, machinery from different vendors, and distinct processes (Raffin et al., 2022). Heterogeneity in the OT Layer presents significant challenges in industrial environments due to diverse industrial communication protocols, devices, and data formats. In modern factories and industrial settings, machines need to communicate with each other to share data and commands. Machines use different protocols (standardized rules) to communicate. Some examples of widely used protocols are OPC UA (open platform communications unified architecture), MODBUS, BLE (Bluetooth Low Energy) etc. (Cha et al., 2023). This diversity, while offering specialized functionality, creates integration barriers. Each protocol has different data formats and structures. It is a challenge to integrate these different data structures and use them as the data source to build automated pipelines for MLOps.

*Challenge 2 – data-related challenges:* data-related challenges like data quality, availability issues, pre-processing, and feature engineering – dealing with complex data from multiple sources, inconsistent data format types, and imbalanced data are some of the common data-related challenges (Bachinger et al., 2024; Hegedus & Varga, 2023; Raffin et al., 2022; Venanzi et al., 2023). In specific domains like additive

**Table 4.** Category 2 – challenges codes abbreviations.

Abbreviations	Codes
C-DDF	Challenges related to inconsistent data format
C-DM	Challenges related to data management
C-DQ	Challenges related to data quality or quantity
C-DOE	Deploying ML solutions on the edge or challenges related to edge implementation
C-HME	Integration challenges due to heterogeneity in manufacturing processes
C-HOT	Integration challenges due to heterogeneity in the OT layer, which includes industrial control systems, devices, and protocols
C-IET	Interpretability, explainability, and Trust challenges
C-NTC	Non-Technical challenges related to cost
C-NTO	Non-Technical challenges related to organizational aspects and work culture
C-NTS	Non-Technical challenges related to a lack of required skills
C-O	Challenges related to operations
C-S	Scalability issues
C-SDT	Scaling/integration challenges of Digital Twin Systems

**Table 5.** Comparison of authors and challenges.

Authors, year	C-DDF	C-DM	C-DOE	C-DQ	C-H	C-IET	C-NT	C-O	C-S	C-SDT
Raffin et al. (2022)	✓	✓		✓	✓		✓	✓		
van Bruggen et al. (2024)										✓
Bachinger et al. (2024)				✓		✓			✓	✓
Cha et al. (2023)					✓					
Venanzi et al. (2023)				✓			✓			
Safdar et al. (2024)		✓					✓			
Chatterjee et al. (2022)		✓		✓					✓	
Alvaro Luis et al. (2023)			✓							
Martel et al. (2020)							✓	✓		
Hegedus and Varga (2023)			✓	✓		✓				
Chatterjee et al. (2022)				✓						
Antonini et al. (2022)			✓							

manufacturing, generating sufficient quality and quantity of data needed to train ML models for industrial AM printers is challenging (Safdar et al., 2024). Unlike areas such as commerce and marketing, where data is available abundantly, the manufacturing sector suffers from a lack of production data, especially in manufacturing processes with infrequent operations (Chatterjee et al., 2022a). The key challenges in implementing MLOps in industrial environments like vacuum pumping stem from the limited availability of ground-truth data, which restricts the ability to train and test machine learning models effectively. This scarcity of data increases the risk of overfitting and makes it difficult to ensure robustness in real-world scenarios (Chatterjee et al., 2022b). Data management difficulties, including handling both structured and unstructured data formats, further complicate model training and performance.

*Challenge 3 – operations-related challenges:* the success of machine learning models is closely tied to their training environment, and any alterations in production processes or manual adjustments can impact their performance, requiring frequent updates and continuous monitoring (Raffin et al., 2022). Model operations in manufacturing setup face challenges around performance degradation over time through model drift and data drift, requiring both technical and business-level monitoring of deployed models, and the need to continuously train and retrain models as new data becomes available to maintain optimal performance (Martel et al., 2020).

*Challenge 4 – scalability-related challenges:* in industrial applications, automated ML systems face the challenge of the need to scale quickly to handle sudden surges in data volume. Additionally, the heterogeneous landscape of ML frameworks, where different software environments must be managed flexibly, poses a scalability challenge. These scalability demands are particularly crucial in industrial settings where continuous processing of manufacturing data is important (Bachinger et al., 2024). Industries frequently scale up, and when they add new hardware, it may differ from existing machinery. The rapid adoption of new hardware into existing processes creates a scalability issue (Chatterjee et al., 2022a).

The scalability of digital twin systems presents significant challenges when implementing machine learning capabilities. As the number of digital twin instances grows, managing multiple ML workflows becomes increasingly complex. This complexity leads to escalating resource requirements, rising infrastructure costs, and the challenge of handling multiple data streams and ML models across numerous digital twins operating simultaneously (van Bruggen et al., 2024).

*Challenge 5 – edge implementation-related challenges:* the deployment of machine learning models at the network edge introduces several critical technical barriers in industrial IoT settings. Inherent resource constraints of edge devices create fundamental barriers for ML implementation (Alvaro Luis et al., 2023). Unlike cloud environments, edge devices operate with minimal processing capabilities, restricted memory, and limited network bandwidth. The industrial context amplifies these challenges, as edge devices are often basic embedded systems lacking conventional computing interfaces. Organizations face additional complexities when attempting to scale ML deployments across multiple edge devices, particularly due to varying hardware architectures and the absence of robust development tools tailored for industrial IoT environments (Alvaro Luis et al., 2023). Adding on, Hegedus and Varga (2023) also highlight the limited capabilities of edge AI platforms compared to cloud platforms. Antonini et al. (2022) discuss about the ‘far edge devices’ which are more resource-constrained and power-efficient, operating at the very extremity of

the network and highlight the challenges regarding implementing ML capabilities on them due to their severe resource constraints and inability to use standard MLOps tools.

*Challenge 6 – challenges related to interpretability, explainability and trust:* the ability to understand and explain how AI models make decisions is fundamental for building confidence in their outputs, especially in manufacturing environments where facilities run for many years and where prediction errors could have serious consequences (Bachinger et al., 2024). Though interpretability and explainability enable informed model validation and model selection by domain experts, this is a time-consuming initiative (Bachinger et al., 2024). Building transparent model architectures and integrating domain knowledge with contextual explanations is essential for improving the interpretability, explainability, and trustworthiness of AI solutions in industrial processes (Hegedus & Varga, 2023). These aspects are critical to the success of an effective MLOps framework.

*Challenge 7 – non-technical challenges:* some of the non-technical challenges include the most obvious aspects related to costs, organizational challenges, and a shortage of skills. Cost is a critical concern to justify the application of ML or ML operations related to infrastructure (Raffin et al., 2022). This challenge is even more profound for SMEs with limited budgets (Venanzi et al., 2023). SMEs also lack sufficient know-how to properly exploit new technologies. Enterprise AI adoption faces organizational challenges from unclear data science role definitions, cultural conflicts between data science and the software team's workflows, practices, and tools (Martel et al., 2020). Since MLOps lies at the intersection of IT software engineering and data science, addressing this challenge is vital for successfully operationalizing AI solutions in enterprise environments.

### **4.3. Recommendations for integrating MLOps methodology**

Several interesting solutions are proposed in this section for the challenges raised in the reviewed articles. These recommendations are based on identifying challenge patterns across various reviewed articles and linking them to architectural characteristics, represented by feature codes, which have the potential to serve as solutions. These solutions are aimed at overcoming both technical and non-technical challenges in implementing MLOps in manufacturing, creating a supportive environment for machine learning to enhance manufacturing processes.

*Recommendation for challenge 1 – protocol level integration in OT layer:* inconsistent data format is a widely acknowledged challenge in an industrial setup. Among the many reasons for this, the diversity of machinery and equipment, coupled with varied communication protocols, often leads to incompatibility with standardized Industry 4.0 frameworks. Addressing this issue requires adopting unified data standards, middleware solutions, or edge devices capable of translating diverse protocols into a common language for enhanced compatibility. Raffin et al. (2022) propose an event-driven microservice approach with standardized messaging using a lightweight message broker (MQTT) at the edge. This architecture transforms data into standardized Industry 4.0 conform events and uses standardized interface services for sensors and PLCs. Cha et al. (2023) mention that protocol integration is achieved through driver mapper development, MQTT protocol implementation, and integration with middle edge nodes. Hegedus and Varga (2023) discuss an architecture that makes use of a Data platform layer and a digital twin layer. While the former is a dedicated data platform that handles extract, transform, and load operations to standardize data coming from diverse sources, the latter is an intermediary layer that can help standardize the interaction with heterogeneous physical systems by providing a unified virtual representation.

*Recommendation for challenge 2 – incorporating data management components in architecture:* the reviewed articles in their own application context describe methods to deal with data-related issues. It is beneficial to have components in MLOps architecture that specifically deal with data-related issues, if any. One study mentions using DAQS for specific sensors and quality checks during acquisition and various efficient data management initiatives in the architecture, like feature stores, caching mechanisms, and multi-modal data storage systems (Raffin et al., 2022). The architecture used by Venanzi et al. (2023) highlights an ETL module called MOMIS (Mediator Environment for Multiple Information Sources) (Magnotta et al., 2018) component in DEEPMON, which enriches and standardizes raw data coming from the field according to configurable data models or metadata. In the context of Digital twins, van Bruggen et al. (2024) discuss that the proposed TDTA aggregates data from multiple digital twin instances of the

same type and supports with more data for training. Adding to this, Hegedus and Varga (2023) also acknowledge digital twin integration to help generate additional training data. Cha et al., 2023 suggest shape-constrained regression to overcome data scarcity by incorporating domain expert knowledge into the training phase. The architecture referred to in Chatterjee et al. (2022a) adopts a modular approach to data QA, where each component of data QA is an ML software in the MLOps architecture. This architecture proposes organizing QA dimensions in a connected graph structure with parent-child relationships. The architecture also incorporates a long-term storage solution, which is a message forwarding system for cloud services, where data is forwarded from the network to a delivery and collection microservice to the cloud API layer for persistent storage.

*Recommendation for challenge 3- Effective maintenance and monitoring throughout the stages of ML life cycle:* referring to Section 4.1, the ‘Monitoring and Maintenance’ theme is consistently present across all architectures in one form or another. A robust and effective model monitoring system is essential to mitigate model drift, ensuring timely and effective maintenance (which could be simple and automated retraining in case of model maintenance) of the various components involved in the MLOps architecture and enhancing operational capacity. Coupled with smooth version control, as referred to in the ‘version control’ theme of Section 4.1, these practices are crucial for maintaining model performance and ensuring traceability. However, this monitoring and maintenance system does not always need to be fully automated; incorporating human-in-the-loop approaches may be more suitable, depending on the specific use case.

*Recommendation for challenge 4- Modular approach:* a modular approach helps address scalability challenges in industrial applications by breaking down complex systems into smaller, independent, and reusable components. Modular architectures align well with containerization and orchestration tools (like Dockers and Kubernetes), enabling efficient scaling and deployment across cloud-edge environments. The architecture presented by Bustamante et al. (2023) emphasizes modularity through a structured pipeline design that separates machine learning operations into distinct components. This approach divides the ML workflow into independent modules for data handling, model training, and deployment, each operating as a self-contained unit. By leveraging containerization, the system enables flexible deployment while maintaining separation of concerns between the base framework, service scripts, and trained models. This modular structure aims to support reusability and maintainability across different ML implementations in industrial IoT environments. Bachinger et al. (2024) stress the modular approach for scalability through the architecture design using task distribution library and a LINQ-query-based approach. The architecture discussed by van Bruggen et al. (2024) addresses scaling challenges in digital twin systems through the MLOps counterpart concept that enables unique ML workflows for each digital twin and standardizes multiple digital twin instances of the same type using the ‘Type Digital Twin Aggregate’ (TDTA) concept.

*Recommendation for challenge 5: opting for containerized microservices and cloud-edge hybrid architectures:* Edge AI implementation faces two key challenges: the resource constraints of edge devices, such as limited computation and memory, and the difficulty of running ML artifacts, particularly for complex lifecycle stages like training and inference. Referring back to the theme ‘microservices and containerization’ in section 4.1, container-based deployment enables running of ML services on resource-limited devices, Alvaro Luis et al. (2023). It is a well-known fact that the usage of containerized Docker images is recommended for lightweight implementation; however, the architecture discussed by Bustamante et al. (2023) uses the Docker BuildX plugin to build images for multiple architectures, which enables deployment across heterogeneous edge devices. Chatterjee et al. (2022a) highlight the different network architectures based on the processing requirements and resource constraints, like End-to-end on-device processing, Data processing on edge and fog nodes, IoT-edge/fog-cloud architecture. The authors of this article note that these architectures can be combined to form a hybrid network architecture for automated ML applications, allowing organizations to balance resource constraints with processing needs. The architecture proposed by Chatterjee et al. (2022b) implements microservices on edge nodes and divides functionality between edge and cloud. Cha et al. (2023) propose their architecture, which addresses edge deployment challenges through KubeEdge and EdgeMesh. KubeEdge is an open-source platform that extends Kubernetes capabilities to edge computing scenarios. EdgeMesh works alongside KubeEdge to handle network communication (edge-to-edge, cloud-to-edge) aspects of the MLOps platform. The

architecture called the Tiny MLOps introduced by Antonini et al. (2022) addresses the resource constraint issue in the far edge devices by adapting the models to lightweight algorithms for the limited hardware, and uses these for basic detection in the embedded devices and uses complex models for verification in the gateway.

*Recommendation for challenge 6: human-in-the-loop approach:* although MLOps can be perceived as a technically oriented solution, the involvement of experts pertaining to its application domain is vital. Raffin et al. (2022) and Cha et al., 2023 discuss the opportunities to build trust with the ‘human-in-the-loop’ HITL) approach with the help of active learning and validation through golden samples (obtaining input from skilled labor as a benchmark). MLOps architectures reviewed in this work possess monitoring architectural elements, which are important aspects in building interpretability and trust from the end users. The architectures discussed by Raffin et al. (Hegedus & Varga, 2023; Raffin et al., 2022; Venanzi et al., 2023) propose comprehensive monitoring services to detect discrepancies such as process drift, model drift, and data drift. Dashboards that continuously monitor and visualize drifts can enhance transparency as to why retraining is required. Bachinger et al. (2024), van Bruggen et al. (2024), Cha et al. (2023); Chatterjee et al. (2022b) specifically emphasized incorporating domain knowledge into the design of the solution, helping domain experts’ enhanced interpretability. The Olympics Model proposed by Hegedus and Varga (2023) includes validation and verification (VV) components that address explainability and trust by discussing procedures and methods for responsible usage, human acceptance, decision control allowances, safety, and time-criticality.

*Recommendation for challenge 7 – non-technical solutions:* Raffin et al. (2022) suggest clear abstraction boundaries through the DDD approach and a structured division of responsibilities through bounded contexts to mitigate organizational challenges for MLOps adoption. Adding on, they suggest investments in efficient and optimized training pipelines to mitigate initial costs. The platform proposed by Cha et al. (2023) allows selective configuration of Kubeflow and the data lake based on the manufacturing site’s scale and construction stage. This modular approach helps the SMEs implement only what they need, leading to a reduction in initial costs. This platform is specifically designed to be applicable to most small and medium-sized enterprises operating various types of semi- automated and automated facilities. The architecture proposed by Venanzi et al. (2023) attempts to bridge the gap between the IT and OT layers by providing a clear separation of responsibilities between layers and a standardized communication protocol. This separation and clear definition are vital for organizational integration. Martel et al. (2020) stress the cross-functional Teams Structure, which combines traditional roles (Business Analysts, Data Scientists, Data Engineers) with additional roles (Software Developers/ML Engineers, DevOps Engineers, Architects etc.) for enterprise deployment. They emphasize a platform which Offers flexibility in choosing between cloud platforms and on-premise solutions and propose a two-track solution approach (code first and low-code approach) based on the organization’s capabilities to mitigate existing skill requirement-related challenges.

Challenges and recommendations mapping: Figure 6 maps the above-stated challenges to recommended architectures that can address them, serving as a reference tool for practitioners. The recommendations are derived from the existing literature and aligned to architectures that best align with addressing specific challenges based on the methodology referred to in Figure 5 in Section 3.2.

*Literature support for proposed MLOps solutions:* existing literature suggests the use of specialized components in the MLOps process to integrate these elements within the OT layer (Silva et al., 2022). Additionally, it emphasizes the importance of incorporating dedicated data management features into the system’s design to tackle data-centric issues. Peer-reviewed studies indicate that constant retraining and updating of ML models are required to maintain performance, which in turn helps to combat operations-related challenges (Raffin et al., 2022; Raffin et al., 2022). By separating various phases of the machine learning pipeline, adopting a modular strategy facilitates the independent expansion of specific components, thus avoiding disruptions to the system as a whole (Fragueiro et al., 2024). Segmenting applications into microservices improves the scalability and manageability of individual components without affecting the system as a whole. This is especially beneficial in edge computing, where resource constraints demand efficient management (Chen et al., 2024; Lin Chen et al., 2020; Ning et al., 2022). Container deployment, with its lightweight and modular characteristics, is optimal for resource-limited edge devices, enhancing resource efficiency and speeding up deployment (Feng et al., 2024; Lin Chen et al., 2020). The synergy

between edge and cloud resources is crucial for managing latency and bandwidth effectively (Chen et al., 2024), which are vital for addressing edge implementation challenges within an MLOps architecture. Integrating expert insights during model training cycles aids in bias reduction, promoting fairness and ethics in algorithms (Kang et al., 2021). HITL strategies facilitate ongoing model refinement through human feedback, fostering trust as models adapt to align with user needs (Retzlaff et al., 2024).

Drawing on the themes and features of the architectures discussed in this article, combined with an in-depth analysis of each architecture from the review and supported by existing literature insights quoted above, the authors recommend referring to the corresponding MLOps architectures in Figure 6, which effectively discusses and addresses the identified challenges. These mappings provide a structured approach to tackling the multifaceted challenges encountered in MLOps, ensuring that the architectures proposed are well supported by current research and practical applications.

## 5. Discussion

Scholarly work increasingly highlights the significance of AI and ML in modern business practices, as more industries begin to leverage these technologies to address practical challenges (Lawrence et al., 2022). By integrating AI and ML into manufacturing, companies unlock new opportunities for optimizing production, reducing costs, improving quality, and enhancing decision-making processes. The research community recognizes the challenges in deploying ML models to production systems and emphasizes the importance of finding solutions to overcome these obstacles, aiming to increase the number of ML models successfully implemented in practice.

Given recent developments, solving industrial problems using AI/ML under experimental conditions can no longer be seen as innovative unless it is successfully implemented in production systems and adds value to the organization. The limited literature on the implementation, validation, and testing of MLOps methodologies in manufacturing, along with the fact that all the reviewed articles in this article are from 2021 onwards, indicates that this field is relatively new and under-researched. This emphasizes the need for more in-depth studies and scholarly efforts to advance and refine practices in this area.

Key features of MLOps contribute to the deployment of ML models (RQ1): the literature review aims to identify the key features of both implemented and proposed MLOps architectures across various application areas in manufacturing. To address RQ1, the review explores different sectors within manufacturing, such as additive manufacturing, digital twin technologies, and vision-based inspection systems, among others. The selected review articles fall into two categories: some present MLOps architectures specifically designed for particular manufacturing sectors, such as additive manufacturing, Industry 5.0, or industrial vacuum pump manufacturing; others propose general architectures or best practices and incorporate relevant use cases in their discussions. The authors of this article take a bottom-up approach, examining individual manufacturing application areas and identifying common themes and insights related to MLOps architectural best practices for the successful implementation of AI and ML in industry. Adding on, this article focuses on MLOps architectures used in real-world scenarios. Four architectures from reviewed articles have been tested and validated in practical applications. Another six are specifically tailored for manufacturing contexts and are on the verge of testing, which is the immediate next step in the respective studies. Selecting architectures that have been empirically tested in real-world settings as an inclusion criterion strengthens the findings on the key characteristics, challenges, and recommendations for successful MLOps adoption.

*Integrating DevOps practices and adopting End-to-End lifecycle management:* the significance of adapting traditional CI/CD pipelines for machine learning workflows is a common thread that runs through all the analyzed articles, even though many address it implicitly, while others emphasize this practice directly (Alvaro Luis et al., 2023; Martel et al., 2020; van Bruggen et al., 2024). This widespread, if sometimes unstated, incorporation of CI/CD principles shows their important role in modern ML systems. Although CI/CD and end-to-end lifecycle management is the backbone of the MLOps framework, it is worth noting that architecture exhibits a high degree of automation throughout the end-to-end lifecycle while still requiring manual intervention and oversight at various stages (Alvaro Luis et al., 2023; Bachinger et al., 2024; Chatterjee et al., 2022a; Raffin et al., 2022; Venanzi et al., 2023). While MLOps is designed to automate and streamline different phases of the ML lifecycle, manual supervision and

assistance remain necessary at certain stages, like initial model development setup, compliance and governance, quality assurance configuration, infrastructure setup, management, business rules and domain logic, and in some cases model maintenance strategies and approval. The level of automation in some architectures is intentionally designed with human oversight and intervention to maintain quality control while still gaining the efficiency benefits of automation for the stable, validated pipelines (van Bruggen et al., 2024). This aligns with typical MLOps practices where full automation is not always desirable, especially for complex ML systems that require domain expertise and careful validation.

*The role and importance of edge devices, containerization, and microservices:* edge computing and IoT integration are fundamental components of modern MLOps architectures, particularly in industrial applications where real-time data processing and decision-making are essential. The integration of edge computing and IoT devices into MLOps frameworks enables the deployment, management, and monitoring of machine learning models directly on devices at the edge of the network, thus reducing latency, managing bandwidth efficiently, and strengthening data privacy (Antonini et al., 2022). In most of the architectures examined in these articles, edge devices serve as the target for deploying machine learning models in time-sensitive scenarios (Alvaro Luis et al., 2023; Antonini et al., 2022; Chatterjee et al., 2022a, 2022b; Hegedus & Varga, 2023; Cha et al., 2023; Raffin et al., 2022; Safdar et al., 2024; Venanzi et al., 2023). The role of edge devices in enabling real-time decision-making is crucial for industrial applications like predictive maintenance and real-time quality control, where delays could lead to operational failures and inefficiencies (Venanzi et al., 2023). Deploying models on edge devices reduces the need to transmit large amounts of data to the cloud while keeping sensitive industrial data local, fostering data privacy.

However, edge devices come with limitations, including minimal computational power and memory constraints, making it difficult to run complex machine learning models like neural networks (Antonini et al., 2022). Additionally, maintaining deployed ML models on edge devices is challenging due to the need for continuous updates, performance monitoring, and adapting to changing operating environments.

To address these limitations, frameworks such as TinyMLOps, federated learning, and hybrid edge-cloud approaches have been proposed (Antonini et al., 2022; Grzesik & Mrozek, 2024). These frameworks distribute processing between the edge and cloud, allowing edge devices to handle real-time inference while offloading more resource-intensive tasks like model retraining to the cloud. Specifically, approaches like TinyMLOps provide customized solutions designed for resource-constrained environments, enabling efficient model deployment even on devices with severe computational limitations.

The use of microservices architecture, containerization, and orchestration enhances the scalability and flexibility of edge-based MLOps. Containerization plays a critical role by encapsulating ML models and their dependencies, ensuring consistency across various environments, while microservices enable modular deployment, allowing individual components to be updated or scaled independently (Pautasso et al., 2017). Running containerized microservices on edge devices combines the benefits of microservices architecture with edge computing, creating an efficient and scalable system for model deployment and bringing computation closer to the data sources. Leveraging these technologies to optimize the deployment and management of ML models across different infrastructures is a game-changer for MLOps architecture. There is a need for research and development in hybrid architecture for MLOps to incorporate both cloud computing for time-insensitive tasks and edge computing for time-sensitive tasks. The synergy between MLOps, containerization, and microservices is crucial but still requires more research to unlock its full potential. However, optimizing how these components work together in resource-limited settings like the edge remains a challenge. Adding on, with the rise of Industry 4.0, edge devices have become commonplace in many industries, often with easy access to cloud connectivity. This availability presents an opportunity to focus research on architectures that leverage the integration of microservices, containerization, and MLOps, alongside hybrid cloud edges. By doing so, we can accelerate the deployment and realization of ML/AI models in industrial applications, further enhancing operational efficiency and scalability.

*Importance of monitoring and maintenance of ML model:* model maintenance and monitoring are critical components of the MLOps framework, ensuring that machine learning models remain relevant, reliable, and trustworthy in production environments. The reviewed articles collectively emphasize the importance of continuous monitoring, addressing model drift, integrating CI/CD pipelines, and incorporating explainability into machine learning models.

The model monitoring systems presented in the architectures reviewed vary in the degree of automation. Some of the architectures are fully automated, while others support automated pipelines but still rely on human intervention and validation when drifts are detected. For instance, the architectures proposed feature robust, real-time performance monitors that trigger notifications and alerts in case of model performance degradation (Alvaro Luis et al., 2023; Bachinger et al., 2024; Hegedus & Varga, 2023). However, retraining must be manually initiated, requiring human input to define new parameters, adjust rules, and conduct fine-tuning and validation. This highlights the point that full automation may not always be ideal, as maintaining safety, security, accuracy, and control often necessitates human oversight.

Meanwhile, the architecture discussed by Chatterjee et al. (2022b) implements periodic model retraining on a fixed schedule, occurring every few months. A different approach is demonstrated in another architecture proposed by Cha et al. (2023) which not only monitors model performance in isolation but also tracks the manufacturing process itself. This enables automatic retraining when changes in the manufacturing process or environment occur. Further research is needed to explore optimal strategies to balance automation with human oversight.

MLOps implementation challenges and recommendations (RQ2): RQ2 examines the challenges in implementing ML models in manufacturing using MLOps. Through a systematic literature review of 12 articles and thematic analysis, we identified seven distinct challenge categories in industrial MLOps. Our analysis further mapped these challenges to specific architectural solutions (Figure 6), revealing that many challenges concentrate in particular MLOps phases depending on automation maturity and infrastructure characteristics, a relationship not systematically documented in prior work.

Our empirical findings both align with and extend existing MLOps literature. The data-related challenges we identified, particularly the infrastructure requirements for real-time data collection and preprocessing on factory floors, resonate with Kreuzberger et al. (2023) data management concerns, though our work specifically emphasizes constraints unique to manufacturing environments that differ substantially from cloud-based deployments. Our systematic review corroborates this emphasis: Narayanappa and Amrit (2024) revealed data challenges appeared in nine of ten reviewed papers, representing the most frequently cited challenge category. We extend these findings by documenting how data versioning challenges manifest distinctly in resource-constrained industrial settings, where Bodor et al. (2023) note that difficulties in managing Jupyter notebook versions, tracking generated artifacts, and maintaining pipeline versions across iterative development cycles become particularly acute. Integration challenges due to heterogeneity emerged as a cross-cutting theme affecting multiple MLOps activities. While Diaz-de-Arcaya et al. (2024) discuss general deployment complexities, our framework explicitly maps how heterogeneity manifests differently across embedded systems versus IT environments in manufacturing contexts. Faubel and Schmid (2024) multiple case study corroborates these findings, noting that even companies using similar tool combinations face substantial deployment variability based on the industry sector, company size, and resource availability. Their work highlights persistent technical challenges in standardized data access across diverse industrial settings and integration issues with legacy IT systems. Our contribution lies in systematically categorizing these heterogeneity challenges and mapping them to specific architectural recommendations.

Operation-related challenges, particularly monitoring, versioning, and model maintenance, represent critical barriers to production deployment. Our analysis reveals practical trade-offs between comprehensive model versioning and resource constraints on edge devices, adding specificity to the reproducibility concerns discussed by Lima et al. (2022). This finding is supported by Kolar Narayanappa and Amrit's (2024) concept matrix, which identified model issues (scalability, accuracy, versioning, monitoring) in eight of ten papers, making it the second most common challenge after data issues. Significantly, Faubel and Schmid (2024) identified a critical operational gap: companies often lack accurate metrics to detect malfunctions or trigger model retraining, with one interviewee stating they conduct monitoring 'at a very basic level' without comprehensive explainability tools or automated retraining triggers. This gap between conceptual MLOps frameworks and actual industrial practice represents a key finding of our work.

Scalability-related challenges intensify as organizations transition from prototype to production deployments. Faubel and Schmid (2024) describe infrastructure management complexities and resource requirements that significantly impact cost and deployment decisions. Bodor et al. (2023) report that entities may

require 6 to 18 months to deploy a single ML model to production, with scalability concerns contributing substantially to these timelines.

Edge implementation challenges are particularly pronounced in Industry 4.0 contexts, representing a manufacturing-specific concern inadequately addressed in the general MLOps literature. Faubel and Schmid (2024) identify deployment difficulties related to containerization on legacy edge devices, noting runtime performance constraints and tensions between containerized deployment benefits and hardware limitations. Bodor et al. (2023) emphasize the emergence of TinyML paradigms to embed ML solutions in resource-constrained microcontroller-based devices, noting that while ‘ML requires significant power to operate,’ Industry 4.0 deployments increasingly demand processing capabilities on minimally sized equipment. Our framework explicitly addresses this tension by mapping edge implementation challenges to specific architectural patterns combining cloud computing for resource-intensive tasks with edge computing for real-time requirements.

Challenges related to interpretability, explainability, and trust represent critical yet under-addressed barriers to production deployment. Faubel and Schmid (2024) explicitly identify ‘Model and Explainability’ as a key challenge domain, noting that while model monitoring is common practice, explainability solutions remain far less prevalent in case study companies. One interviewee articulated the challenge directly: ‘The main thing is how can we make our black box models more explainable so we can actually show them why the model made such a decision and if it actually was correct or not.’ Kolar Narayanappa and Amrit (2024) found that transparency deficits and difficulties explaining model outcomes to non-technical stakeholders recurred throughout their interviews, particularly for models deployed in high-stakes industrial decision-making contexts. Our work advances this discussion by explicitly mapping human-in-the-loop architectural patterns to interpretability challenges, providing concrete implementation guidance absent from prior literature.

Non-technical challenges, encompassing organizational, business, and human factors, emerged as having the highest impact on MLOps adoption in manufacturing. While John et al. (2021) discuss maturity models and Testi et al. (2022) categorize pipeline types, our framework provides more granular mapping of these challenges to specific MLOps activities, revealing that barriers concentrate in particular phases depending on automation degree and infrastructure maturity. Narayanappa and Amrit's (2024) grounded theory analysis supports this finding, showing organizational challenges had the highest code density (23 codes) across four dimensions, Organizational, Technical, Operational, and Business, with specific issues including human resource and skillset limitations, user engagement and resistance, slow processes, and collaboration barriers. Bodor et al. (2023) emphasize that successful ML projects require coordination between analysts, data scientists, data engineers, front-end engineers, and production engineers to ensure a safe transition from exploration to production environments. Extending these findings, Faubel and Schmid (2024) identified human-machine interaction challenges in high-risk scenarios where defining boundaries between automated and human decision-making becomes problematic, particularly when operators make physical plant changes that invalidate model assumptions. Our finding that CI/CD testing requires ML-specific adaptations complements testing challenges identified in existing literature, while our emphasis on the optional nature of certain activities (such as automated retraining) reflects practical industry conservatism underrepresented in theoretically oriented frameworks. Faubel et al. (2023), Faubel and Schmid (2024), and Faubel et al. (2025) found that while companies maintained well-defined deployment processes and used similar open-source and commercial tool combinations, automation and deployment remained challenging, with one interviewee noting that ‘a lot of things may change and you never know if it's really a problem of the model itself, or it's more a problem of something of the underlying data.’ This uncertainty about root causes, model degradation versus data issues, highlights the complexity of operationalizing continuous training. Bodor et al. (2023) emphasize that continuous training through automated ML training pipelines and continuous monitoring for model health represent properties distinguishing MLOps from traditional DevOps, yet both areas show significant gaps between conceptual frameworks and industry practice.

Thus, our work synthesizes and contextualizes existing challenges within a unified Industry 4.0-specific MLOps activity model while identifying several previously underexplored tensions between automation ambitions and operational realities. Our systematic challenge-to-architecture mapping (Figure 6) provides practitioners with a structured reference tool for selecting appropriate MLOps architectures based on their

specific manufacturing context and challenges, advancing both theoretical understanding and practical implementation of MLOps in industrial environments.

### 5.1. Limitations

This study provides valuable insights into industrial MLOps architectures and implementation challenges, but a few limitations should be considered when interpreting and applying the findings. To begin with, this review analyzed 12 studies selected through strict inclusion criteria focused on manufacturing-specific MLOps implementations with detailed architectural specifications and evidence of real-world deployment or clear implementation readiness. While this quality-focused approach prioritizes practical relevance and empirical grounding, the limited sample size restricts statistical generalizability and may not capture the full diversity of MLOps approaches across different manufacturing contexts. This strict set of inclusion criteria focusing on manufacturing or maintenance contexts with detailed MLOps architectures, while ensuring relevance, may have excluded interesting insights from adjacent industrial domains. The definition of 'implementation-ready' involved subjective judgment, where borderline cases between conceptual frameworks and deployment-ready architectures might be classified differently by other reviewers. However, this quality gate helped filter out purely conceptual work, ensuring that recommended architectures have demonstrated feasibility for industrial deployment.

Although Scopus provides comprehensive coverage of major academic publishers, including IEEE, ACM, Springer, and Elsevier, reliance on Scopus as the only primary database may have resulted in gaps. Specifically, recently published papers not yet indexed, preprints and gray literature from industry sources, conference proceedings with delayed indexing, and technical reports from manufacturing organizations may have been excluded. On the positive side, focusing on Scopus-indexed publications ensures that we have a baseline of peer-reviewed papers with quality and academic rigor, reducing the risk of including unvalidated claims or promotional material often found in gray literature. Furthermore, although thematic analysis had systematic procedures, a few subjective decisions could have influenced the findings. The coding process was interpretive in identifying architectural features and challenges, where the boundaries between the overlapping categories required subjective interpretation. The definition of 'implementation-ready' involved subjective judgment, where borderline cases between conceptual frameworks and deployment-ready architectures might be classified differently by other reviewers. Nevertheless, this quality gate helped filter out purely conceptual work, ensuring that recommended architectures have demonstrated feasibility for industrial deployment.

The reviewed architectures and challenges span various manufacturing domains, including additive manufacturing, predictive maintenance, quality control, and vacuum pumping systems, each with distinct operational characteristics. The general applicability of findings to manufacturing environments with unique regulatory requirements, safety-critical systems, or extremely low-tolerance quality standards needs further work to establish. Stressing more on the generalizability of the mappings, the distinction between discrete versus continuous manufacturing, batch versus flow production, or high-volume versus high-mix environments is not systematically addressed, raising questions about how recommendations should be adapted across these different process types. However, this diversity actually shows that the architectural patterns we identified work across different types of manufacturing, though each implementation will need adjustments for its specific setting.

These limitations suggest that findings should be interpreted as providing directional guidance and structured frameworks for MLOps adoption rather than prescriptive solutions guaranteed to succeed universally. Practitioners should adapt recommendations based on their specific organizational, technical, and operational contexts, ideally supplementing these findings with pilot implementations, iterative refinement, and continuous learning from their own deployment experiences.

## 6. Conclusion

This study aimed to identify the key features of MLOps architectures that facilitate the deployment of machine learning models in a manufacturing context, along with the challenges encountered during implementation, and how these architectures address some of the specific challenges. The results revealed

that the general theme of key features of the reviewed architectures that makes MLOps a suitable approach for addressing the deployment challenges of ML models in dynamic production environments are: its foundation derived with DevOps practices, efficient management of the end-to-end ML lifecycle, edge computing and IoT integration, the use of digital twins and cyber-physical systems, continuous monitoring and maintenance, and the adoption of microservices, containerization, and robust version control. The perceived solution of MLOps to address the deployment challenges of ML models faces its own obstacles in dynamic industrial settings. This article identifies seven key challenges based on the challenges discussed in the review and outlines how these challenges could be addressed by drawing design inspiration from the recommended architectures.

This study offers valuable theoretical contributions by advancing research on effective solutions and practices for implementing ML models in manufacturing. While current literature often focuses on the development of ML solutions, it tends to treat deployment and maintenance as separate challenges. However, the key insight for practitioners is that the development, deployment, and maintenance of ML models are interconnected and should be integrated into the planning phase from the beginning of any ML project. A holistic approach, such as MLOps, which considers the entire lifecycle of ML models, is crucial. There is a growing need for further research on how this comprehensive framework can be tailored specifically in the manufacturing sector. This research offers significant practical contributions by providing manufacturing practitioners with a structured reference tool that maps specific MLOps implementation challenges to recommended architectural solutions. The challenge-architecture mapping serves as a practical decision-making guide, enabling organizations to identify and adopt appropriate MLOps architectures based on their specific manufacturing context and challenges. This practical framework reduces the trial-and-error typically associated with MLOps adoption in manufacturing settings, potentially accelerating implementation timelines and improving success rates for AI/ML initiatives in industrial environments.

Future work will empirically investigate the implementation and construction of MLOps architectures, focusing on the most critical features for effective implementation, as identified in this study, and addressing the challenges encountered during the process. In a broader research sense, future research needs to focus on developing standardized architectures and best practices for implementing MLOps in manufacturing, improving the adoption of MLOps practices across different manufacturing sectors, and investigating how MLOps architectures can be scaled efficiently for large-scale manufacturing operations.

## Acknowledgements

The authors would like to thank the Advanced and Innovative Digitalization Program funded by VINNOVA for their funding of the research project TPdM-Trustworthy Predictive Maintenance (Grant No. 2022-01710). This study has been conducted within the Production Area of Advance at the Chalmers University of Technology.

## Author contributions

CRediT: **Mohan Rajashekarappa**: Conceptualization, Data curation, Formal analysis, Investigation, Methodology, Software, Validation, Visualization, Writing – original draft, Writing – review & editing; **Ebru Turanoglu Bekar**: Conceptualization, Funding acquisition, Project administration, Supervision, Writing – original draft, Writing – review & editing; **Alexander Karlsson**: Conceptualization, Funding acquisition, Project administration, Supervision, Validation, Writing – original draft, Writing – review & editing; **Jon Bokrantz**: Conceptualization, Supervision, Writing – original draft, Writing – review & editing; **Mukund Subramaniyan**: Conceptualization, Writing – original draft, Writing – review & editing; **Anders Skoogh**: Conceptualization, Funding acquisition, Project administration, Supervision, Writing – original draft, Writing – review & editing.

## Disclosure statement

No potential conflict of interest was reported by the author(s).

## Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT 4o in order to proofread and enhance readability. After using this tool, the authors reviewed and edited the content and take full responsibility.

## References

- Amou Najafabadi, F. (2024). Reference architecture of mlops workflows. *ECSCA 2024: European Conference on Software Architecture*. Springer. <https://doi.org/10.1007/978-3-031-7124-36>
- Andrew Tamburri, D. (2020). Sustainable MLOps: Trends and Challenges. *IEEE*. <https://doi.org/10.1109/SYNASC51798.2020.00015>
- Antonini, M., Pincheira, M., Vecchio, M., & Antonelli, F. (2022). Tiny-MLOps: a framework for orchestrating ML applications at the far edge of IoT systems. *IEEE*. <https://doi.org/10.1109/EAIS51927.2022.9787703>
- Diaz-de-Arcaya, J., Torre-Bastida, A. I., Zarate, G., Minon, R., & Almeida, A. (2024). A joint study of the challenges, opportunities, and roadmap of mlops and aiops: A systematic survey. *ACM Computing Surveys*, 56(4), 1–30. <https://doi.org/10.1145/3625289>
- Bachinger, F., Zenisek, J., & Affenzeller, M. (2024, November 22–24). Automated machine learning for industrial applications - challenges and opportunities. *5th International Conference on Industry 4.0 and Smart Manufacturing, ISM 2023*, (pp. 1701–1710). Lisbon, Portugal: Elsevier. <https://doi.org/10.1016/j.procs.2024.01.168>
- Bayram, F., & Ahmed, B. S. (2025). Towards trustworthy machine learning in production: an overview of the robustness in mlops approach. *ACM Computing Surveys*, 57(5), 1–35. <https://doi.org/10.1145/3708497>
- Bodor, A., Hnida, M., & Najima, D. (2023). Mlops: overview of current state and future directions. *7th International Conference on Smart City Applications (SCA 2022)*, (pp. 156–165). Castelo Branco, Portugal: Springer, Cham. <https://doi.org/10.1007/978-3-031-26852-6>
- Bustamante, A. L., Patricio, M. A., Berlanga, A., & Molina, J. M. (2023). Seamless transition from machine learning on the cloud to industrial edge devices with thinger.io. *IEEE Internet of Things Journal*, 10(18), 16548–16563. <https://doi.org/10.1109/JIOT.2023.3268771>
- Cha, J.-H., Heung-gyun, J., Seung-woo, H., Dong-chul, K., Jung-hun, O., Seok-hee, H., & Byeong-ju, P. (2023, July 23–28). Development of MLOps Platform Based on Power Source Analysis for Considering Manufacturing Environment Changes in Real-Time Processes. *25th International Conference on Human-Computer Interaction (HCII 2023)*, (pp. 224–236). Copenhagen, Denmark: Springer, Cham. <https://doi.org/10.1007/978-3-031-35572-1>
- Chakraborty, A., Das, S., & Gary, K. A. (2025). Machine Learning Operations: A Mapping Study. *CSCE 2024 (World Congress in Computer Science, Computer Engineering & Applied Computing)*, (pp. 3–21). Springer, Cham. [https://doi.org/10.1007/978-3-031-86644-9\\_1](https://doi.org/10.1007/978-3-031-86644-9_1)
- Chatterjee, A., Ahmed, B. S., Hallin, E., & Engman, A. (2022a). Quality assurance in mlops setting: An industrial perspective, volume 3362
- Chatterjee, A., Ahmed, B. S., Hallin, E., & Engman, A. (2022b). November 14–18). Testing of machine learning models with limited samples: an industrial vacuum pumping application. *Proceedings of the 30th ACM Joint European Software Engineering Conference and Symposium on the Foundations of Software Engineering (ESEC/FSE 2022)*, (pp. 1280–1290). Singapore: Association for Computing Machinery (ACM). <https://doi.org/10.1145/3540250.3558943>
- Chen, N., Toosi, A. N., Javadi, B., Alqahtani, D., Aslanpour, M. S., & Xu, M. (2024). An empirical study on edge-to-cloud continuum for smart applications: performance, design patterns, and key factors. *IEEE International Conference on Edge Computing and Communications (EDGE)*, *IEEE*. <https://doi.org/10.1109/EDGE62653.2024.00011>
- Chou, P.-N. (2012). A comparison study of impact factor in web of science and scopus databases for engineering education and educational technology journals. *Issues in Informing Science and Information Technology*, 9, 187–194. <https://doi.org/10.28945/1615>
- Schröer, C., Kruse, F., & Marx Gómez, J. (2021). A systematic literature review on applying crisp-dm process model. *CENTERIS / ProjMAN / HCist 2020*, (pp. 526–534, Vol. 181). Elsevier Procedia Computer Science. <https://doi.org/10.1016/j.procs.2021.01.199>
- Faubel, L., & Schmid, K. (2024). MLOps: A Multiple Case Study in Industry 4.0. *IEEE 29th International Conference on Emerging Technologies and Factory Automation (ETFA)*, (pp. 1–8). *IEEE*. <https://doi.org/10.1109/ETFA61755.2024.10711136>
- Faubel, L., Schmid, K., & Eichelberger, H. (2023). Mlops challenges in industry. *SN Computer Science*, 4, 0. <https://doi.org/10.1007/s42979-023-02282-2>
- Faubel, L., Woudsma, T., Klopper, K., Eichelberger, H., Buelow, F., Schmid, K., Ghezaljeheidan, A., Theodorou, A., Meth-nani, L., Theodorou, A., & Bang, M. (2025). Mlops for cyberphysical production systems: challenges and solutions. *IEEE Software*, 42, 65–73. <https://doi.org/10.1109/MS.2024.3441101>
- Feng, Y., Shen, S., Wang, X., Xiang, Q., Xu, H., Xu, C., & Wang, W. (2024). BREAK: A Holistic Approach for Efficient Container Deployment among Edge Clouds. *IEEE International Conference on Computer Communications (IEEE*

- INFOCOM 2024), (pp. 1491–1500). Vancouver, BC, Canada: IEEE. <https://doi.org/10.1109/INFOCOM52122.2024.10621084>
- Fragueiro, F. G., Martín, D. G., López, A. B., Rial, A. A., Tranchero, J. O., Lorenzo, B. C., Montenegro, J. M. F., & Muin˜os-Landin, S. (2024). An integrated active learning framework for the deployment of machine learning models for defect detection in manufacturing environments. In *Lecture Notes in Mechanical Engineering*, (pp. 3–14). Springer, Cham. [https://doi.org/10.1007/978-3-031-57496-2\\_1](https://doi.org/10.1007/978-3-031-57496-2_1)
- Garrone, A., Minisi, S., Oneto, L., Dambra, C., Borinato, M., Sanetti, P., Vignola, G., Papa, F., Mazzino, N., & Anguita, D. (2023). Simple non regressive informed machine learning model for prescriptive maintenance of track circuits in a subway environment. *International Conference on System-Integrated Intelligence*, (pp. 74–83, Vol. 546). Springer, Cham. [https://doi.org/10.1007/978-3-031-16281-7\\_8](https://doi.org/10.1007/978-3-031-16281-7_8)
- Grzesik, P., & Mrozek, D. (2024). Combining machine learning and edge computing: opportunities, challenges, platforms, frameworks, and use cases. *Electronics*, 13(640), 1–26. <https://doi.org/10.3390/electronics13030640>
- Gulshat A, A., Bauyrzhan S, A., Gulnur A, T., & Timur, I. (2024). Application of machine learning algorithms in digital twin monitoring systems: an overview of approaches, methods, and prospects, IEEE. <https://doi.org/10.1109/ICNGN63705.2024.10871832>
- Hegedus, C., & Varga, P. (2023). Tailoring MLOps techniques for industry 5.0 needs. *19th International Conference on Network and Service Management (CNSM)*, IEEE. <https://doi.org/10.23919/CNSM59352.2023.10327814>
- Heymann, H., Mende, H., Frye, M., & Schmitt, R. H. (2023). Assessment framework for deployability of machine learning models in production. *Procedia CIRP*, 118, 32–37. <https://doi.org/10.1016/j.procir.2023.06.007>
- Heymann, H., Kies, A. D., Frye, M., Schmitt, R. H., & Boza, A. (2022). Guideline for deployment of machine learning models for predictive quality in production. *Procedia CIRP*, 107, 815–820. <https://doi.org/10.1016/j.procir.2022.05.068>
- John, M. M., Olsson, H. H., & Bosch, J. (2021). Towards MLOps: A Framework and Maturity Model. *47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, (pp. 334–341). <https://doi.org/10.1109/SEAA53835.2021.00050>
- Kang, Y., Chiu, Y.-W., Lin, M.-Y., Su, F.-Y., & Huang, S. T. (2021). Towards model-informed precision dosing with expert-in-the-loop machine learning. 342–347. <https://doi.org/10.1109/IRI51335.2021.00053>
- Kolar Narayanappa, A., & Amrit, C. (2024). An Analysis of the Barriers Preventing the Implementation of MLOps, (pp. 101–114). Springer, Cham. [https://doi.org/10.1007/978-3-031-50188-3\\_10](https://doi.org/10.1007/978-3-031-50188-3_10)
- Kreuzberger, D., Kuhl, N., & Hirschl, S. (2023). Machine learning operations (mlops): overview, definition, and architecture. *IEEE Access*, 11, 31866–31879. <https://doi.org/10.1109/ACCESS.2023.3262138>
- Larsson, L. (2017). Characteristics of production innovation. Department of Business Administration, Technology and Social Sciences, Division of Innovation and Design, Luleå University of Technology.
- Lawrence, N. D., Paleyes, A., & Urma, R.-G. (2022). Challenges in deploying machine learning: A survey of case studies. *ACM Computing Surveys*, 55(6), 114:1–114:29. <https://doi.org/10.1145/3533378>
- Lima, A., Monteiro, L., Paula, A., & Furtado, C. (2022). Mlops: practices, maturity models, roles, tools, and challenges - a systematic literature review. <https://doi.org/10.5220/0010997300003179>
- Lin Chen, J., Liaqat, D., Gabel, M., & De Lara, E. (2020). Poster: an accelerator for fast container-based applications deployment on the edge 175–177. <https://doi.org/10.1109/SEC50012.2020.00027>
- Moher, D., Liberati, A., Tetzlaff, J., Altman, D. G., Antes, G., Atkins, D., Barbour, V., Barrowman, N., Berlin, J. A., Clark, J., Clarke, M., Cook, D., D’Amico, R., Deeks, J. J., Devereaux, P. J., Dickersin, K., Egger, M., Ernst, E., Gotzsche, P. C., ... Tugwell, P. (2009). Preferred reporting items for systematic reviews and meta-analyses: the prisma statement. *PLoS Medicine*, 6(7), <https://doi.org/10.1371/journal.pmed.1000097>
- Magnotta, L., Gagliardelli, L., Simonini, G., Orsini, M., & Bergamaschi, S. (2018). Momis dashboard: A powerful data analytics tool for industry 4.0. In *Proceedings of the International Conference on Knowledge-Based and Intelligent Information & Engineering Systems*, 1074–1081. <https://doi.org/10.3233/978-1-61499-898-3-1074>
- Martel, Y., Roßmann, A., Sultanow, E., Weiß, O., Wissel, M., Pelzel, F., & Seßler, M. (2020). Software architecture best practices for enterprise artificial intelligence. *volume P*, 307, 165–181.
- Mayr, A., Kießkalt, D., Meiners, M., Lutz, B., Sch˜afer, F., Seidel, R., Selmaier, A., Fuchs, J., Metzner, M., Blank, A., & Franke, J. (2019). Machine learning in production – potentials, challenges and exemplary applications. *Procedia CIRP*, 86, 49–54. <https://doi.org/10.1016/j.procir.2020.01.035>
- Mboweni, T., Masombuka, T., & Dongmo, C. (2022). A systematic review of machine learning devops. <https://doi.org/10.1109/ICECET55527.2022.9872968>
- Ning, L., Yusong, T., Xiaochuan, W., Bao, L., & Jun, L. (2022). Score: A resource- efficient microservice orchestration model based on spectral clustering in edge computing, (pp. 186–202). Springer, Cham. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. [https://doi.org/10.1007/978-3-031-20984-0\\_13](https://doi.org/10.1007/978-3-031-20984-0_13)
- Ouzzani, M., Hammady, H., Fedorowicz, Z., & Elmagarmid, A. (2016). Rayyan—a web and mobile app for systematic reviews. *Systematic Reviews*, 5(1), 210. <https://doi.org/10.1186/s13643-016-0384-4>
- Pautasso, C., Zimmermann, O., Amundsen, M., Lewis, J., & Josuttis, N. (2017). Microservices in practice, part 1: reality check and service design. *IEEE Software*, 34, 91–98. <https://doi.org/10.1109/MS.2017.24>

- Raffin, T., Reichenstein, T., Klier, D., & Franke, A. (2022). Qualitative assessment of the impact of manufacturing-specific influences on machine learning operations. *Procedia CIRP*, 115, 136–141. <https://doi.org/10.1016/j.procir.2022.10.063>
- Raffin, T., Reichenstein, T., Werner, J., & Franke, A. (2022). A reference architecture for the operationalization of machine learning models in manufacturing. *Procedia CIRP*, 115, 130–135. <https://doi.org/10.1016/j.procir.2022.10.062>
- Rani, F., Chollet, N., Vogt, L., & Urbas, L. (2024). Industrial edge mlops: overview and challenges. *Computer Aided Chemical Engineering* (53, pp. 3019–3024. <https://doi.org/10.1016/B978-0-443-28824-1.50504-4>
- Retzlaff, C. O., Das, S., Wayllace, C., Mousavi, P., Afshari, M., Yang, T., Saranti, A., Angerschmid, A., Taylor, M. E., & Holzinger, A. (2024). Human-in-the-loop reinforcement learning: A survey and position on requirements, challenges, and opportunities. *Journal of Artificial Intelligence Research*, 79, 359–415. <https://doi.org/10.1613/jair.1.15348>
- Safdar, M., Paul, P. P., Lamouche, G., Wood, G., Zimmermann, M., Hannesen, F., Bescond, C., Wanjara, P., & Zhao, Y. F. (2024). Fundamental requirements of a machine learning operations platform for industrial metal additive manufacturing. *Computers in Industry*, 154, 104037. <https://doi.org/10.1016/j.compind.2023.104037>
- Sasu Makinen, H., Skogstrom, E., & Laaksonen, T. (2021). Mikkonen. Who needs mlops: what data scientists seek to accomplish and how can mlops help? 109–112. <https://doi.org/10.1109/WAIN52551.2021.00024>
- Scharp, K. M., & Sanders, M. L. (2019). What is a theme? Teaching thematic analysis in qualitative communication research methods. *Communication Teacher*, 33(2), 117–121. <https://doi.org/10.1080/17404622.2018.1536794>
- Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., Chaudhary, V., Young, M., Crespo, J.-F., & Dennison, D. (2015). Hidden technical debt in machine learning systems. volume 2015-January, page 2503 – 2511
- Silva, A. S. D., Van, H. M., & Weiss, G. (2022). Implementing a metadata manager for machine learning with the asset administration shell. *volume*, 2022. <https://doi.org/10.1109/ETFA52439.2022.9921671>
- Steidl, M., Felderer, M., & Ramler, R. (2023). The pipeline for the continuous development of artificial intelligence models—current state of research and practice. *Journal of Systems and Software*, 199, 111615. <https://doi.org/10.1016/j.jss.2023.111615>
- Subramanya, R., Sierla, S., & Vyatkin, V. (2022). From devops to mlops: overview and application to electricity market forecasting. *Applied Sciences (Switzerland)*, 12(19), 9851. <https://doi.org/10.3390/app12199851>
- Symeonidis, G., Nerantzis, E., Kazakis, A., & George, A. (2022). Pa- pakostas. Mlops - definitions, tools and challenges 453–460. <https://doi.org/10.1109/CCWC54503.2022.9720902>
- Testi, M., Ballabio, M., Frontoni, E., Iannello, G., Moccia, S., Soda, P., & Vessio, G. (2022). Mlops: A taxonomy and a methodology. *IEEE Access*, 10, 63606–63618. <https://doi.org/10.1109/ACCESS.2022.3181730>
- van Bruggen, A. H., Kruger, K., Basson, A. H., & Grobler, J. (2024). An architecture to integrate digital twins and machine learning operations. *Stud Comput Intell* (Vol. 1136, pp. 3–14). *SCI*. [https://doi.org/10.1007/978-3-031-53445-4\\_1](https://doi.org/10.1007/978-3-031-53445-4_1)
- Venanzi, R., Dahdal, S., Solimando, M., Campioni, L., Cavalucci, A., Govoni, M., Tortonesi, M., Foschini, L., Attana, L., Tellarini, M., & Stefanelli, C. (2023). Enabling adaptive analytics at the edge with the bi-rex big data platform. *Computers in Industry*, 147. <https://doi.org/10.1016/j.compind.2023.103876>
- Woźniak, A. P., Milczarek, M., & Woźniak, J. (2025). And joanna Woźniak. mlops components, tools, process, and metrics: A systematic literature review. *IEEE Access*, 13, 22166–22175. <https://doi.org/10.1109/ACCESS.2025.3534990>
- Wohlin, C. (2014). Guidelines for snowballing in systematic literature studies and a replication in software engineering. *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering* (pp. 1–10). <https://doi.org/10.1145/2601248.2601268>
- Zarour, M., Alzabut, H., & Al-Sarayreh, K. T. (2025). Mlops best practices, challenges and maturity models: A systematic literature review. *Information and Software Technology*, 183, 107733. <https://doi.org/10.1016/j.infsof.2025.107733>