Institute of Software Engineering

University of Stuttgart
Universitätsstraße 38
70569 Stuttgart

Masterarbeit

Exploring Real-World Challenges in MLOps Implementation: A Case Study Approach to Design Effective Data Pipelines

Vidushi Arora

Course of Study:	Computer Science
Examiner:	Prof. Dr. Stefan Wagner
Supervisor:	Markus Haug
Commenced:	November 15, 2023
Completed:	May 15, 2024

Acknowledgement

I extend my heartfelt gratitude to my family and friends for their unwavering support throughout this journey. Their encouragement has been my anchor, driving my determination forward.

I am profoundly thankful to Markus Haug, my supervisor at the university, whose continuous guidance and insightful feedback have significantly shaped this thesis. Also, special appreciation goes to Christoph Caprano, my advisor at the company, whose valuable insights were pivotal in refining my research.

I am also grateful to all the interviewees who generously shared their time and expertise, enriching the study with diverse perspectives. Your insights have been instrumental in shaping the findings of this research. Furthermore, I extend my gratitude to my colleagues for their insights and feedback during the solution development. I also thank all the colleagues who supported and motivated me during this journey, making it not just a successful one but also a pleasant and memorable experience.

I would like to extend heartfelt thanks to Bastian Behrens, Christoph Caprano, and everyone at the partner company Valiton for making this opportunity possible through their support. Additionally, I am sincerely grateful to Dr. Prof. Stefan Wagner for giving me the opportunity to write a thesis at the Institute of Software Engineering and for his guidance during its examination. I am also thankful to Dr. Prof. Stefan Wagner and the other members of Empirical Software Engineering for their valuable feedback during the mid-term presentation.

Lastly, I acknowledge my consistency and persistence throughout this journey, which was crucial for the success of this thesis.

Abstract

With the increasing significance of machine learning (ML) systems across various industries, leveraging Machine Learning Operations (MLOps) to effectively streamline the lifecycle of ML models has become a focal point. Central to the MLOps is the effective management of data, which is a core component of machine learning models. In this context, a data pipeline is utilised, which consists of a series of automated processes for moving and transforming data to be analysed using machine learning systems. However, to stay ahead in the ever-evolving field of Machine Learning Operations (MLOps), it is critical to understand the challenges involved in building efficient and reliable data pipelines. This study investigates these challenges using a comprehensive approach to gain diverse perspectives. It starts with a systematic literature review to identify academic viewpoints, followed by a multiple case study that examines various industry projects through interviews with practitioners. This approach seeks to validate the academic findings in real-world contexts and identify gaps in the existing research.

The study's outcome demonstrates a considerable alignment between the challenges documented in academic literature and those faced by industry professionals in their projects. Regardless, the significance of these challenges varied between the two sectors, and insights from industry practitioners highlighted challenges that academic research had overlooked. Based on these insights, the study further explored version management as a significant challenge that was reflected in the case studies as lacking knowledge and being inadequately addressed. This led to the development of a solution that demonstrates practical applicability to effectively mitigate the challenges and ensure adaptability in real-world settings.

In conclusion, this study presents a detailed and comprehensive analysis of the challenges impacting the MLOps data pipeline lifecycle, highlighting both the consistency and unique aspects of academic and real-world perspectives. The findings will help shape more effective data pipelines that meet real-world needs. Additionally, the study presents a solution to tackle the identified significant challenge, providing valuable insights into its advantages and limitations. It also reflects on the practical implications of this solution and proposes future directions for the field.

Contents

List of Figures

List of Tables

Acronyms

ACM Association for Computing Machinery. 35

AI Artificial Intelligence. 20

AIOps Artificial Intelligence Operations. 21

AWS Amazon Web Services. 53

CI/CD Continuous Integration/ Continuous Deployment. 21

CSV Comma-Separated Values. 31

DevOps Development and Operations. 17

DVC Data Version Control. 61

ECS Elastic Container Service. 53

GDPR General Data Protection Regulation. 55

IEEE Institute of Electrical and Electronics Engineers. 35

ML Machine Learning. 19

MLOps Machine Learning Operations. 5

OLAP Online Analytical Processing. 46

RQ Research Question. 16

SLR Systematic Literature Review. 21

VS Code Visual Studio Code. 77

1 Introduction

The rapid expansion of data availability and technological advancements have fueled the adoption of data-driven processes to make informed decisions in business environments [BM16]. As organisations increasingly depend on intricate data, the need for sophisticated methods to manage and leverage this information has intensified. Machine learning has become a prominent approach for utilising data to foster business innovation [PF13]. As a subset of artificial intelligence, machine learning focuses on developing algorithms that enable systems to learn from data, make decisions, and improve autonomously over time. This capability is particularly valuable in handling large and complex data sets, positioning data as a crucial asset for machine learning systems [EM15]. Nevertheless, integrating machine learning into business operations presents unique challenges compared to traditional software, as these systems combine extensive data processing with advanced models, and this integration demands robust and efficient practices. As a result, Machine learning operations (MLOps) have emerged as a vital discipline for seamlessly integrating machine learning models into production environments. MLOps covers the end-to-end lifecycle of machine learning models, ranging from deploying them to keeping them running smoothly, ensuring they are scalable, maintainable, and comply with organisational and regulatory standards [WKRM22] [CD11]. The performance of machine learning systems is closely tied to the nature of the algorithms they use. These algorithms depend on high-quality and appropriate data, making data management critical throughout the MLOps lifecycle. However, effective data management in real-world scenarios involves navigating a spectrum of complexities, such as managing the significant volume, velocity, and variety of data and adopting MLOps with rapidly evolving tools and practices [RDW+21]. Recognising and understanding these diverse challenges is essential to addressing them and developing a robust data pipeline to support machine learning projects' success.

1.1 Research Goals and Objective

Considering the importance of recognising and understanding the challenges, this thesis aims to explore and provide insights into data-management challenges across different phases of the MLOps lifecycle. This exploration is critical to ensuring an effective data pipeline in a machine learning system. This study aims to comprehensively assess the difficulties mentioned in the existing academic literature and encountered by professionals in the field to identify key data management challenges in machine learning systems. This thesis will also explore the gap between academic literature and industry concerns, highlighting relevant challenges for real-world use cases that will guide further academic research and inform industry practice. Moreover, by leveraging significant insights from industry and academia, this thesis aims to address significant challenges and scenarios identified in industry case studies utilising MLOps principles to enhance machine learning system development.

To achieve these research goals and objectives, we have formulated the following research questions (Research Questions (RQs)) that this study will aim to answer:

RQ1 What are the primary challenges in the lifecycle of data pipelines in machine learning systems?

- **RQ1.1** What are the main challenges regarding the lifecycle of data pipelines highlighted in the academic literature on machine learning topics?
- **RQ1.2** What are the primary challenges regarding the lifecycle of data pipelines encountered in industrial applications?
- **RQ1.3** How do these challenges identified in academic literature **RQ1.1** and encountered in industrial application **RQ1.2** compare?

RQ2 How can MLOps methodologies be implemented to address the identified key challenges and enhance the efficiency of machine learning development?

1.2 Outline

This thesis is organised into several chapters. Chapter 2 offers an overview of MLOps, focusing on its architecture and data management practices. Chapter 3 reviews prior work pertinent to our study. This is followed by Chapter 4, which details the research methodology, including the research questions and methodologies employed, such as systematic literature reviews, interviews, thematic synthesis, and the design science approach. Chapter 5 presents the results, providing insights from the systematic literature review and interviews, and addresses the challenges and solutions to identified problems. The next is Chapter 6, which explores the gaps between industry and academia regarding various challenges, outcomes of the solutions, potential future work, and threats to validity. Finally, Chapter 7 concludes the thesis by summarising our findings and providing an overarching perspective of the study.

2 Background

This chapter will provide more insight into MLOps and a brief overview of the data pipeline that we are examining in this thesis.

MLOps, as illustrated in Figure 2.1, integrates practices and principles from fields of machine learning, Development and Operations (DevOps) and data engineering [TBF+22]. The discussion starts with a brief overview of DevOps and data engineering, which are foundational to MLOps, followed by a detailed examination of MLOps. It includes exploring the need for MLOps, its principles, architecture, and workflow.

Data Engineering

Data engineering encompasses a wide range of practices designed to prepare and manage data efficiently using repetitive and scalable solutions [RDW+21]. These practices include data collection, ingestion, preparation, analysis, storage, monitoring, testing, validation, visualisation, management, and orchestration.

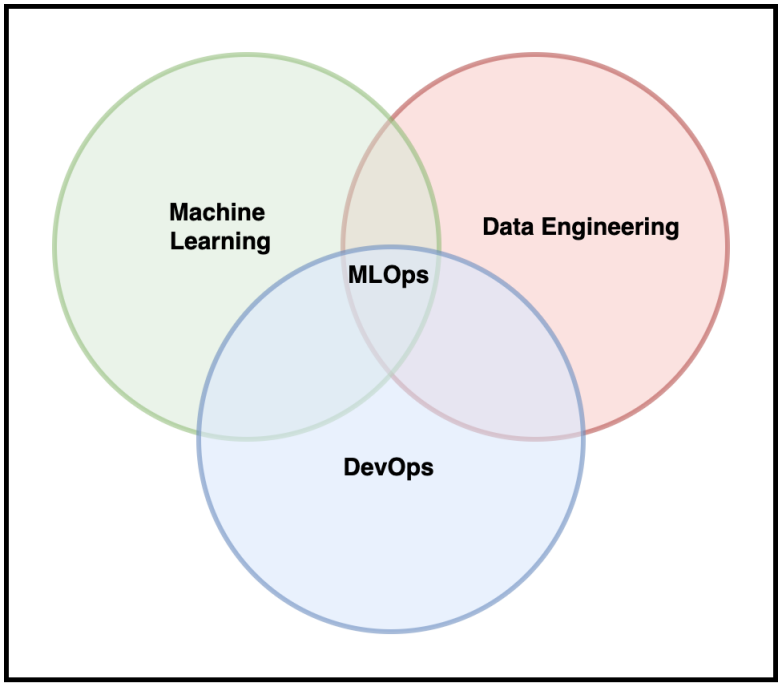

Figure 2.1: Foundations of MLOps [TBF+22]

DevOps

The adoption of DevOps in software engineering systems has earned popularity due to its numerous benefits. These include reduced time-to-market cycles, enhanced collaboration, and improved operations, all of which contribute to higher quality [EAN+17]. DevOps aims to reduce the gap between software development and its operations, thereby shortening the release cycle time [FS17]. This is primarily achieved through processes such as Continuous Integration [KPYL08] and Continuous Deployment [CBA15], which automate the testing, integration, and building of systems with new code changes, ensuring they are ready for deployment and delivery to the production environment. [ZM22] [NKD13]. Each of these processes plays a significant role in ensuring the quality and usability of the data, highlighting the significance of data engineering in the MLOps pipeline.

MLOps

With their exploratory nature, machine learning systems present unique challenges that differ from traditional software systems [SKS21]. These challenges, particularly in model development and deployment, underscore the importance of MLOps. The iterative nature of machine learning experiments, with their varying configurations, can make tracking changes in code, data, parameters, and configurations challenging, potentially hindering reproducibility. Testing and deployment in ML systems must consider both data and models, and monitoring must extend to model performance, not just software system components. Thus, MLOps aims to address these differences and enable the seamless development of machine learning systems [KKH23]. MLOps principles include versioning of data, model and code, CI/CD automation, workflow orchestration, reproducibility, versioning of data, model and code, ML metadata tracking and logging, collaboration, continuous ML training and evaluation, continuous monitoring data, model and feedback [KKH23] [MLO] The processes in MLOps are iterative and incremental and mainly have three phases: design, experimentation or development, and operations [MLO]. Multiple studies have proposed MLOps workflows. However, we will discuss the one proposed by Kreuzberger, D. et al. (2023) [KKH23], shown in Figure 2.2; its comprehensive MLOps architecture and workflow includes roles, interactions and other components. The MLOps workflow details steps such as defining business problems, requirements, and goals to ascertain necessary data types; designing and developing data engineering pipelines for feature preparation; model development for training and evaluating models; and automated ML workflows for model preparation, registration, and deployment for production and model serving [KKH23].

Data, models, and code are major components integrated across the various processes of MLOps [MLO]. Integrating data and model management enhances communication, tracking, and governance of ML artefacts, thereby improving governance, interoperability, and transparency [SKS21].

Product Initialisation

The definition and documentation of business problems provide a significant understanding of the other stages of acquisition, preparing, testing, and monitoring data and models [TBF+22]. This also evolves over time with feedback whenever deemed necessary. After the needs and requirements are defined, the data sources are searched, and data acquisition or collection starts. It involves selecting and analysing initial data [TBF+22].

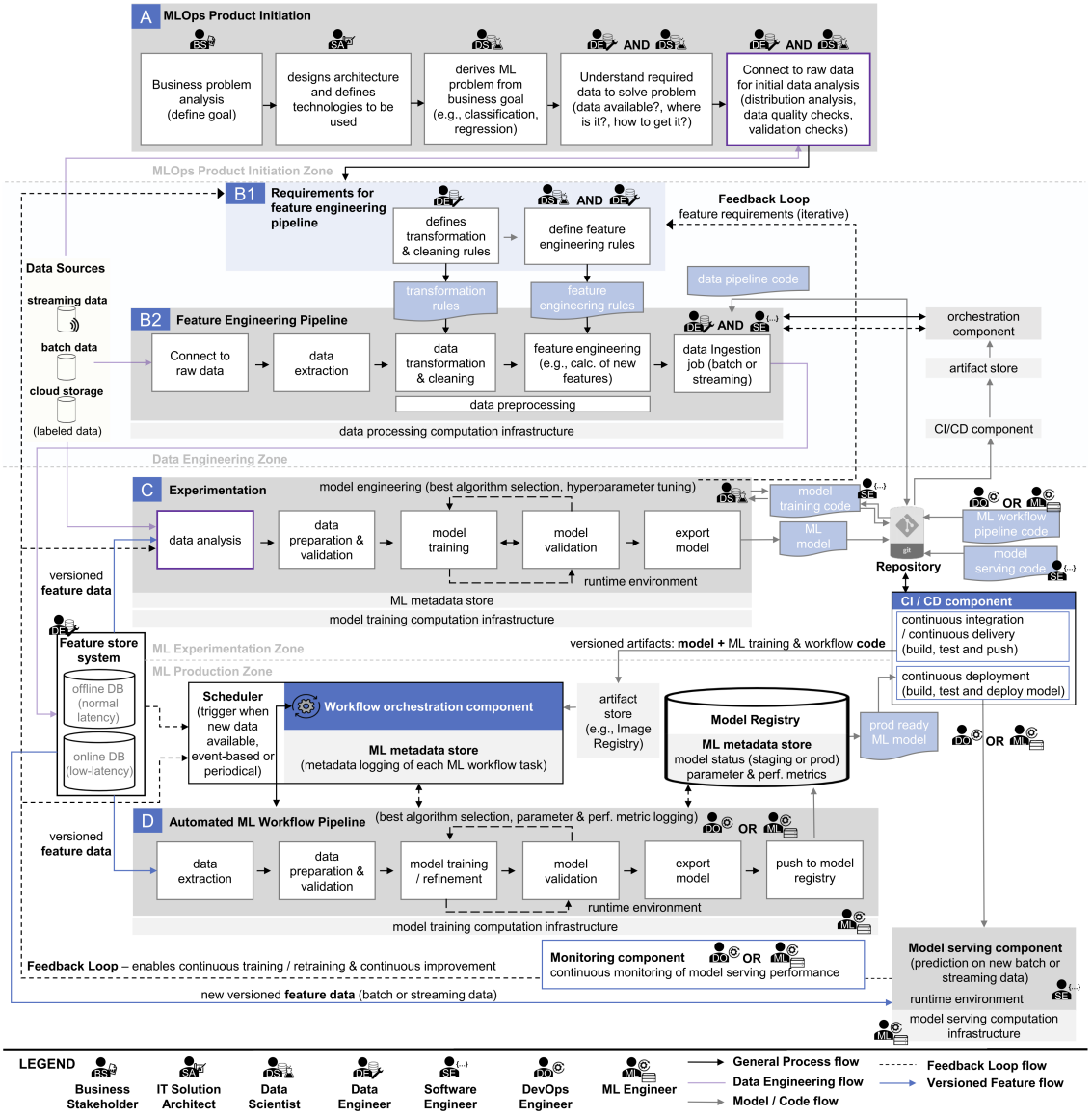

Figure 2.2: MLOps workflow [KKH23]

Data Pipeline

The data pipeline consists of a series of operations that process data [RBOW20]. It is identified as a feature engineering pipeline in the data engineering zone, as shown in Figure X. Data sources, whether streaming or batch, feed into these pipelines [MW15]. The data relevant to the Machine Learning (ML) problem is then ingested, collected, explored, validated, processed, and prepared in a format suitable for ML models [TBF+22] [MLO]. Features are prepared, versioned, and stored in a feature store. Understanding the business use case is essential to define feature requirements, which can be iteratively updated based on feedback to optimise model performance or to meet changes in business requirements [KKH23] [MLWS22]. Data engineering is crucial for machine learning projects, facilitating the automation of data management and enhancing monitoring and fault detection [RBOW20]. This process starts with modelling the pipeline by identifying

data-consuming activities, establishing their sequence, and designing monitoring and storage solutions [RBOW20]. The pipeline is then developed and implemented, involving the construction, cleaning, and transformation of datasets [MLO] [RBOW20], a process that demands significant time and effort.

ML Model Development and Automation

Experimentation is a critical part of ML model development. It uses versioned features to process and validate data further for model training and evaluation. This phase requires multiple experiments, selecting different models and dataset versions, tuning hyperparameters, and configuring to meet business requirements [JOB21]. Along with code versioning and using a shared code repository, this phase also needs versioning of models, data, and other parameters for better collaboration, reproducibility, traceability, and quality [Pul24]. The selected model, code, and data versions are then used to trigger a workflow that automates the ML development steps in an isolated environment, with iterative training to achieve benchmark results and register the successful model in a model registry with the appropriate version and status [JOB21] [KKH23], enabling continuous integration

Scalable pipelines for parallel experiments are developed, including steps for model training and evaluation. Continuous training pipelines are necessary in a production environment to train models at appropriate intervals or when data changes or model performance degrades [SKS21]. Identifying and defining triggers for these pipelines is also a complex task. Model status in the model registry can initiate a continuous deployment pipeline that builds, packages, tests, and deploys the model pipeline to the target environment where model serving occurs [KKH23] [SKS21]. Continuous monitoring is required to evaluate model performance and provide feedback for retraining or improvement in model development. Monitoring encompasses model and data, system performance, and resource utilisation [SKS21] and can also generate explanations and serving logs of the model [APLC22].

Ethical Considerations in MLOps

This workflow, however, does not include components for data privacy and security, ethical and legal requirements, explainability, and sustainability. The MLOps workflow proposed by Testi et al. [TBF+22] includes explainability and sustainability. Explainability in machine learning focuses on understanding the underlying mechanisms of predictions and the rules followed to justify them, enabling stakeholders to comprehend and assess the model's decision-making process [BAWX20]. Sustainability concerns the understanding and minimising of the energy and carbon footprint of activities or products, including the direct and indirect carbon dioxide emissions generated throughout their lifecycle [TBF+22]. The topics of ethics, explainability, and sustainability have gained prominence with the rise of Artificial Intelligence (AI) and ML systems and their real-world influence to protect moral principles, avoid bias, and ensure fairness.

3 Related Work

Numerous review papers on machine learning systems emphasise key challenges across different phases and stages of ML systems. For example, in 2022, Andrei et al. focussed on ML system deployment by reviewing case studies and identifying data management as one of the main stages for deployment [CCL+20].

The paper highlights various technical concerns at each stage that require attention for resolution. Similarly, another study by Kolltveit & Li, from 2022 concentrates on operationalising ML systems and identifies gaps within current tools and infrastructure for operationalising such systems through a Systematic Literature Review (SLR) [KL23]. Although this paper does not explicitly focus on challenges related to the data management stage or tools for its operationalisation, it does reflect on identifying gaps within current MLOps tools. Other review papers focusing on specific aspects of machine learning systems have also brought attention to data-related challenges. For example, Al Alamin and Uddin conducted a Systematic Literature Review (SLR) to determine quality assurance challenges across different phases of ML development [AU21]. The result was challenges mapped to the corresponding ML pipeline stages and phases of ML development. A significant portion of the identified challenges was associated with all stages of the ML pipeline. This study emphasised the prominence of data-related quality assurance challenges across various lifecycle phases of ML systems. Following this, Shivashankar & Martini, 2022 explored maintainability challenges concerning different stages of the ML pipeline, including the data engineering stage [SM22]. They observed that quality issues in other stages could necessitate repetitive maintenance activities in the data engineering and model training stages. We also identified the most recent review papers focusing on challenges in broader systems, including AI and ML systems. For instance, Diaz-de-Arcaya et al. highlighted data management as one of the main challenges in adopting MLOps and Artificial Intelligence Operations (AIOps), with a broad focus on issues like data quality, accessibility, and processing [DTZ+23]. Furthermore, a multi-vocal literature review combined with interviews by Steidl et al. in 2023 underscored challenges in Continuous Integration/ Continuous Deployment (CI/CD) of pipelines, including data handling for AI systems [SFR23].

There are also papers focused on insights from industry, such as Nahar et al., who conducted interviews focusing on collaboration challenges in ML systems, discussing events and associated challenges, including those related to training data negotiation [NZLK22]. Muiruri et al. examined standard practices and challenges across various ML pipeline stages, highlighting data quality and collection challenges due to data source inconsistencies [MLKM21]. They also emphasised gaps in MLOps tools and their adoption.

Several studies focused on understanding the challenges in MLOps or machine learning systems in general, as well as specific aspects of them. These papers repeatedly highlighted data-related challenges across different stages and contexts. Several studies emphasised the need for further research on data management challenges. However, none of the identified papers offered an in-depth review of data-related challenges covering all the stages in MLOps. Moreover, the review papers

gained insights using only academic literature and did not complement the results with those of industry professionals. Our study aims to fill this gap with insights from the academic review and industry case studies. In fact, the results supported the challenges highlighted by these related studies and provided more understanding and context of the priority and relevance of these challenges for practitioners.

Beyond identifying challenges, several papers offer solutions to address these challenges, such as guidelines for implementing MLOps principles [MG22] or proposing frameworks to tackle software engineering issues in ML systems [SV22]. Others evaluate the time and performance implications of using ML tools to construct ML pipelines [ZYD20]. Our study seeks to provide solutions to these challenges from the perspective of data scientists, aiming to bridge the gap in tools and guidance regarding versioning tools and pipelines for improved data and model management during model development.

4 Methodology

This chapter explains how different research methods were employed to achieve the study goals and address the research questions.

For RQ1, which aims to identify and understand data management challenges in academia and industry, different methods were tailored to each sector. A systematic literature review in academia was conducted to collect relevant studies, and the findings were analysed using thematic synthesis to identify underlying themes. This approach is illustrated in Figure 4.1. A case study approach was adopted for industry challenges, involving interviews with practitioners working with data pipelines and machine learning projects. These interviews were also analysed using thematic synthesis, maintaining consistency in analysis across both sectors.

To address RQ2, we leverage the insights derived from the case studies conducted for RQ1. These findings are then applied within a design science approach to identify further and scope the problem and explore and propose solutions.

Each of these methodologies will be explored in detail in the subsequent sections of this chapter.

4.1 Systematic Literature Review

The Systematic Literature Review (SLR) methodology is frequently employed in Software Engineering to gain an in-depth understanding of specific research topics [ZB13]. It leverages existing research and is particularly valued for its ability to yield more reliable findings and identify areas needing further investigation [ZB13]. This methodology is well-suited for addressing RQ1, especially RQ1.1, which focuses on identifying the challenges and their prominence, and RQ1.3, which compares their prominence in the context of industry applications. To systematically identify relevant literature, this study integrates the search strategies proposed by Kitchenham et al. [KC07] and Wohlin [Woh14]. This approach is detailed in Figure 4.1. Further details about the search strategy, study selection, and applied constraints will be discussed in subsequent sections.

Kitchenham's approach to SLR involves an exhaustive search across various databases utilising relevant keywords to identify potential research articles. This method requires the execution of various search queries across multiple databases to find pertinent literature systematically, which can be quite exhausting and time-consuming [KC07]. Conversely, the snowballing method introduced by Wohlin offers a more expedient alternative for locating relevant literature across diverse journals and databases, bypassing the need for individual database searches [Woh14]. This technique initiates with a starting collection of relevant articles to further discover additional literature through forward (looking at papers that cite the initial set) and backward (examining the references within the initial papers) citation tracing. Despite the differences in both of these approaches, they both tend to yield comparable outcomes, as evidenced by the research conducted by Badampudi et al. [BWP15].

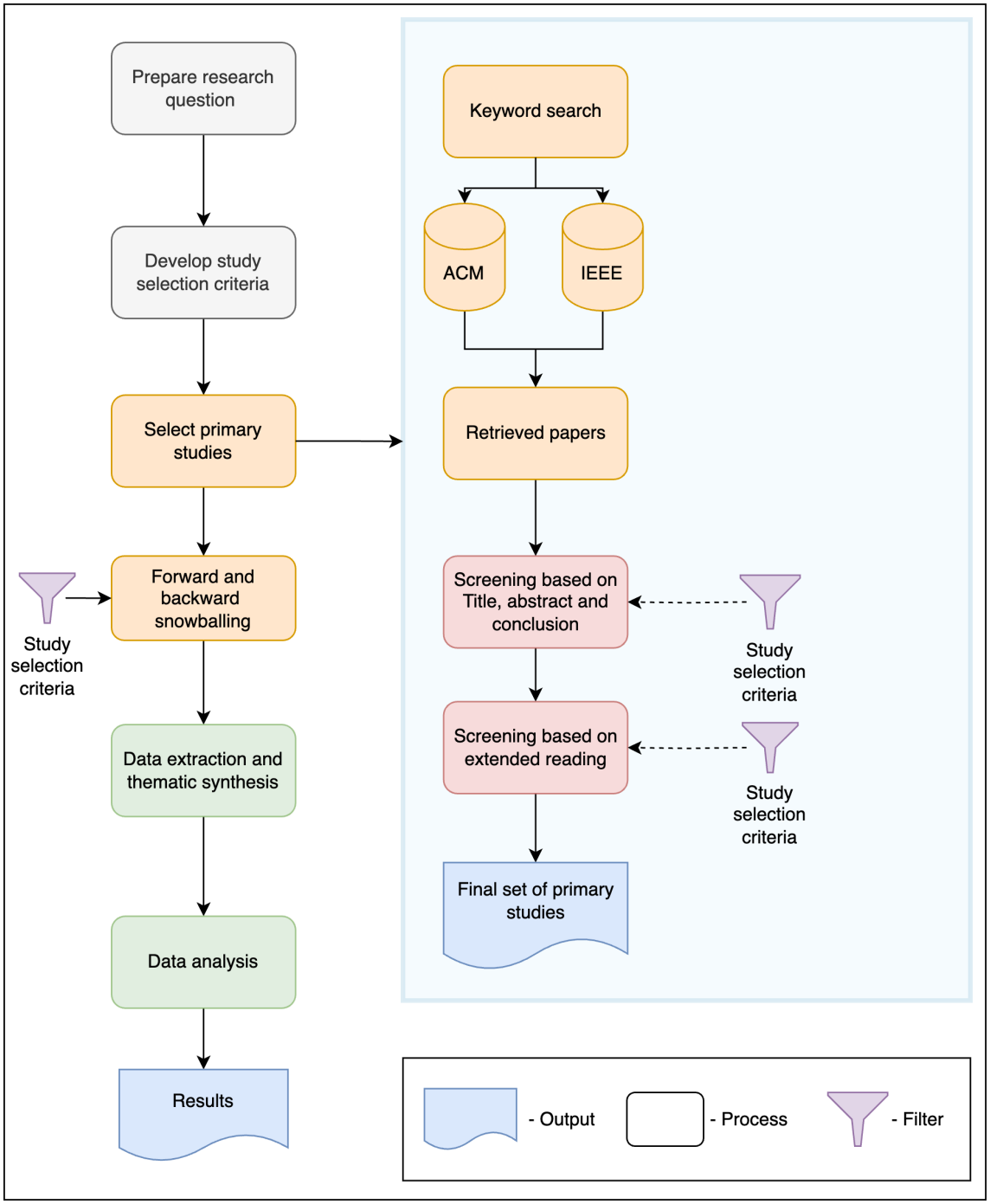

Figure 4.1: Research Methodology for Identifying Data Management Challenges in Academic Studies

4.1.1 Search Strategy Overview

The initial stage of this study involved conducting searches within two popular digital libraries in the field of Computer Science. However, due to the limited number of relevant studies discovered through this initial search and the time constraints of the thesis, the snowballing technique was subsequently employed to broaden the search scope and conserve time that would otherwise be spent on searching additional libraries individually. The papers identified in the initial search phase were the starting point for snowballing. The success of the snowballing approach largely depends on the quality of the initial batch of papers, and selecting studies from two notable libraries helps ensure a diverse and representative foundation for further research [BWP15]. Due to time and resource constraints, the snowballing process was confined to a single iteration. The search string employed the keywords "MLOps" AND "challenge" AND "data" with the following rationale:

MLOps: This keyword narrows the focus to the specialised area of Machine Learning Operations (MLOps), emphasising the automation, management, and scalability of data pipelines essential for the efficacious deployment of machine learning models. The MLOps framework, as outlined in the study, underscores the automation and enhancement of the processes associated with the development, deployment, monitoring, and maintenance of ML systems, including data management as a pivotal component.

Data: The term 'data' is broadly applied to encompass all aspects related to data that could influence and evolve data pipelines. It ensures that all relevant literature is covered.

Challenge: Ensures the inclusion of studies discussing challenges, aligning the search with the thesis's objectives.

The search was confined to publications released between January 1, 2020, and December 31, 2023. The selected timeframe considered the significance of an influential paper that defined the understanding of the MLOps framework authored by John et al., published in 2021 [JOB21]. Also, Google Trends data reveals a rising global interest in MLOps [clo], depicted in Figure 4.2.

4.1.2 Study selection process

The papers obtained from our search strategy were carefully screened and selected according to the study selection criteria outlined in Table 4.1. For inclusion, the studies were required to satisfy all quality assessments and any specified inclusion criteria while not meeting any exclusion criteria. The inclusion criteria were specifically chosen based on their ability to address RQ1.1, focusing on challenges within data pipelines. Specifically, inclusion criterion (IC3) targeted data-related issues in MLOps, recognising that any data-related issues could indirectly or directly affect decisions within the data pipeline. Exclusion criteria were applied to studies with clear language barriers or access restrictions and those falling outside the defined timeline, a factor already discussed in the search strategy overview. These criteria were crucial for snowballing, ensuring the search remained focused on relevant studies during this phase. The quality of the papers was assessed with an emphasis on peer-reviewed work to compensate for our limited experience in evaluating studies independently. The screening process entails systematically evaluating the titles, abstracts, and conclusions of the retrieved papers. Should these sections exhibit ambiguity or provoke

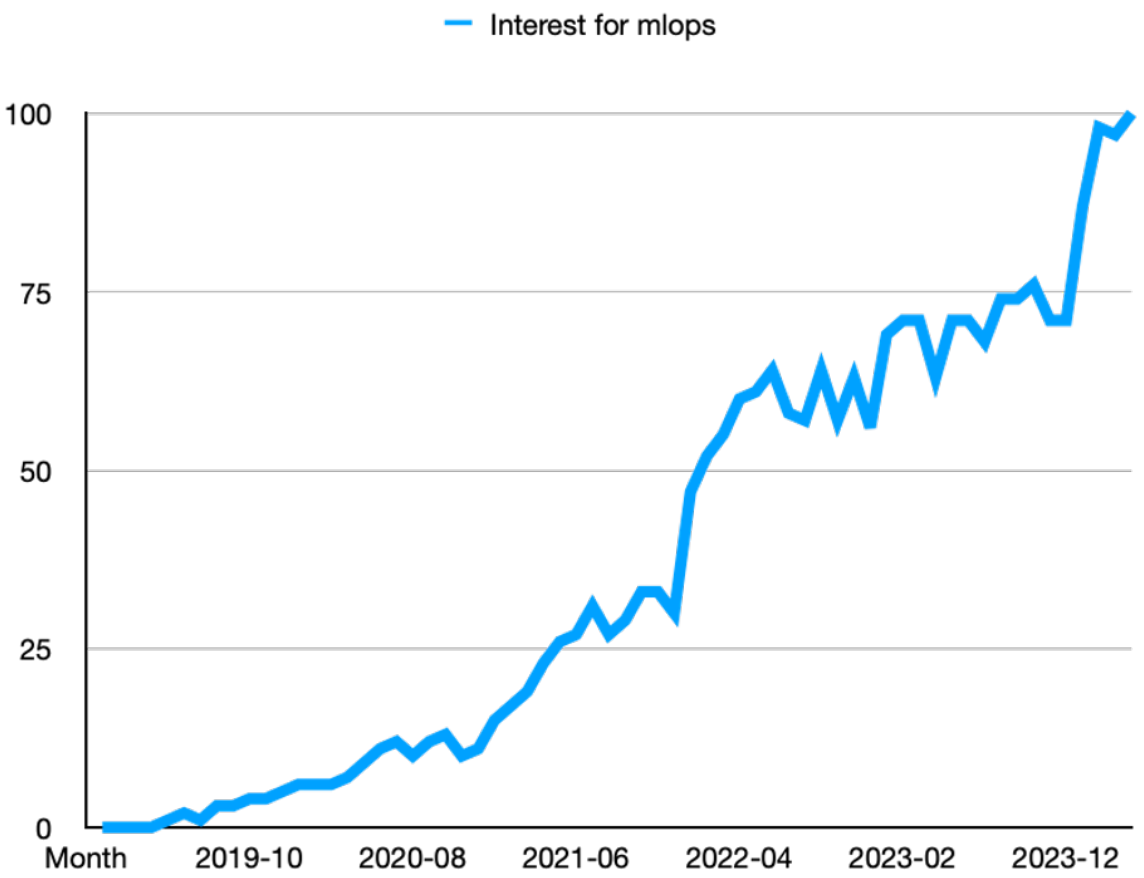

Figure 4.2: Trends in MLOps Interest Over Time, Based on Google Trends Data [clo]

uncertainties, a comprehensive review of the text is undertaken. During the selection phase, review studies are excluded from the analysis to prevent bias from duplicated insights but are included as primary papers in the snowballing phase to identify relevant studies.

4.2 Interview

The interviews were conducted as part of a qualitative research methodology involving a range of professionals affiliated with the partner company of this thesis. These professionals spanned roles such as software engineers, data engineers, data scientists, data architects, and product managers, all of whom have engaged in either data engineering or machine learning projects. The interviews were semi-structured, guided by the template questions with themes identified in a Systematic Literature Review (SLR). The question template is available in the Appendix B. The open-ended questions were used to keep the discussion unbiased, allowing for a broad exploration of topics beyond those identified in the SLR. Before participating, all individuals were briefed on the thesis's objectives and the methods for data handling. They were required to consent formally to their participation, with details outlined in the appendix. The interviews were recorded and transcribed using the Teams platform. The recording was used to correct the automatically generated transcript and ensure the accuracy of the information captured. The transcriptions were anonymised and made available to

	Inclusion Criteria
IC1	Studies addressing challenges associated with data management in the context of machine learning systems.
IC2	Studies exploring challenges related to the development and maintenance of data pipelines.
IC3	Studies delving into challenges concerning data within the framework of MLOps.
	Exclusion Criteria
EC1	Studies that are not in English
EC2	Studies that are not accessible
EC3	Studies that are published before 01.01.2020
	Quality Assessment Criteria
QC1	Studies that are peer reviewed

Table 4.1: Study Selection Criteria

the interviewees for review, allowing them to propose any amendments or retract certain statements, thus upholding ethical research standards. After their approval, the original recordings were deleted, leaving only the amended transcriptions for analysis. This procedure is depicted in Figure 4.3.

4.3 Thematic Synthesis

Thematic synthesis is a qualitative analysis method designed to systematically discover patterns, themes, and relationships within data [CD11]. This method requires comprehensively extracting relevant lines from chosen documents or studies, addressing the research question. The extracted data then undergo systematic coding to highlight significant patterns. This study employs an integrated coding strategy, beginning with provisional codes derived from the research question and existing literature. As the analysis advances, these codes can be refined, or entirely new codes can emerge from the data, providing a rich understanding of established concepts and fresh discoveries. The process unfolds in several stages: Initially, coding maps the extracted data to broader themes that offer insights related to the research question, using existing or newly created codes as needed. These themes are then grouped into higher-level themes, forming new conceptual categories. Finally, synthesis is evaluated for its reliability and applicability. In the analysis process, thematic synthesis first scrutinises studies from a Systematic Literature Review (SLR), employing provisional codes informed by initial exploration of data pipeline and MLOps concepts from the literature. When categorising lines under a specific theme, a challenge code is applied to each study at most once, ensuring that multiple references within the same study are counted as one occurrence to maintain objectivity and avoid over-representation. Subsequently, thematic synthesis is applied to interview transcripts. If applicable, codes from the SLR thematic synthesis are utilised as initial codes and new codes are generated as necessary. This approach facilitates a nuanced comparison of challenges faced in the industry with those outlined in academic literature. When categorising interviewees' statements under challenge codes, a challenge code is acknowledged once per project to avoid bias in results. This methodology also highlights if some interviewees under the same project did not experience particular challenges, providing a comprehensive view of the project landscape.

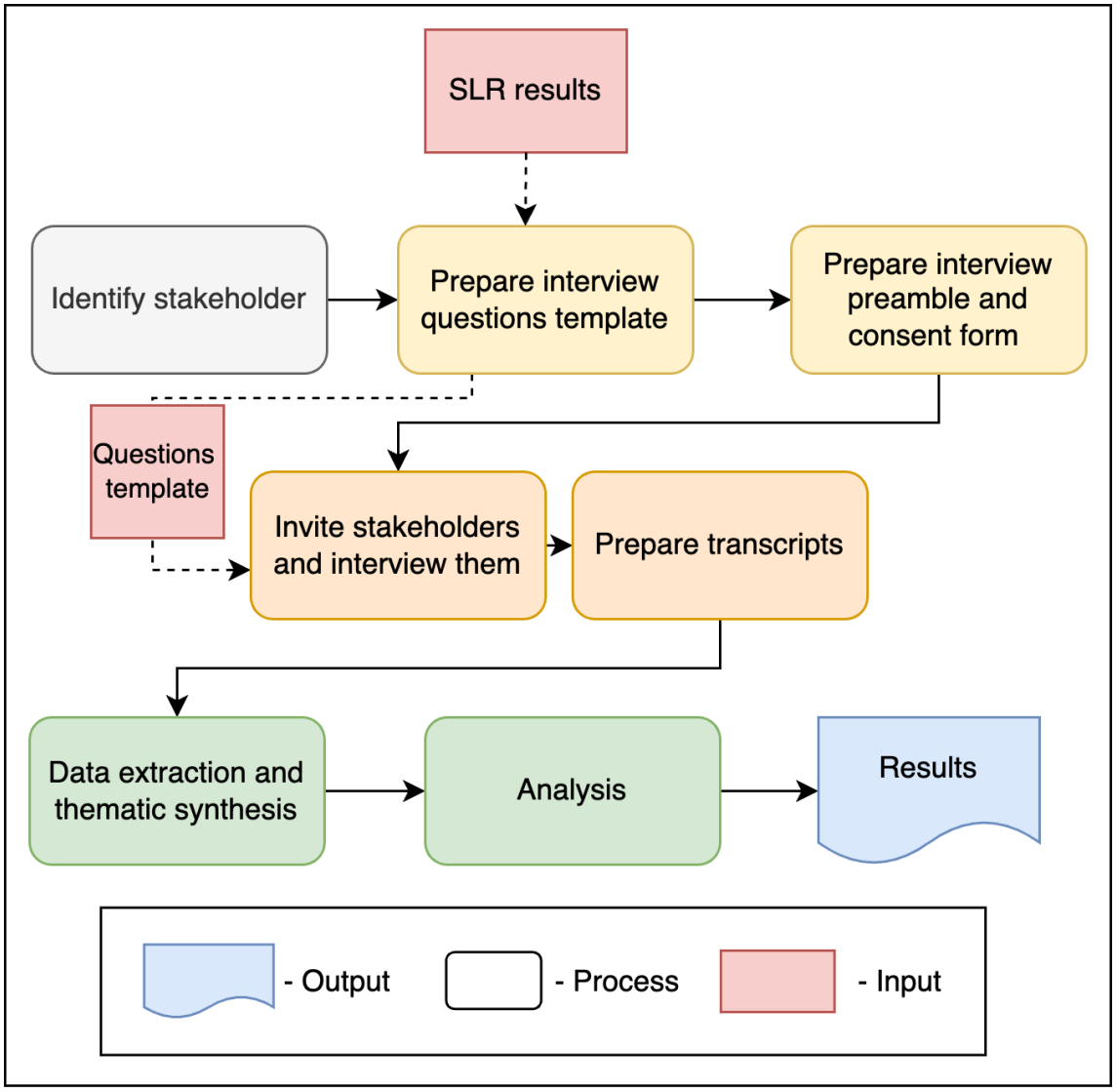

Figure 4.3: Research Methodology for Identifying Data Management Challenges in Industry

4.4 Design Science Approach

This section describes the methodology employed to identify, define, and explore solutions for issues emerging from the thematic synthesis of interviews. The design science approach is adapted to pinpoint problems, conceptualise scenarios, and develop solutions addressing real-world needs [PTRC07]. This method is depicted in Figure 4.4, illustrating the iterative processes of problem identification, scenario conceptualisation, and solution implementation.

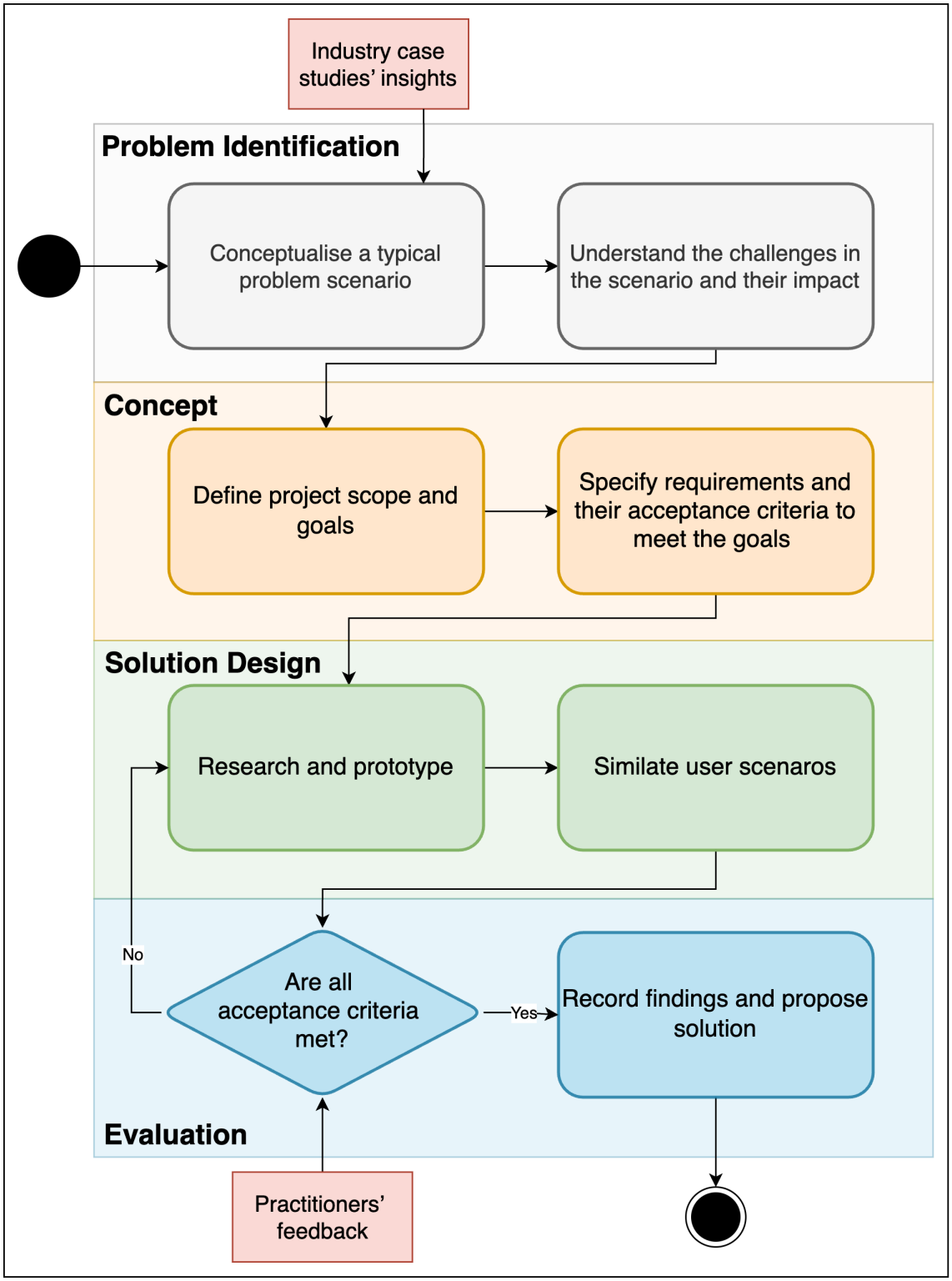

Figure 4.4: Design Science Approach

4.4.1 Problem Identification

Data pipeline operations pose significant challenges in machine learning projects, encompassing design, integration, tool selection, versioning, scalability, and maintenance issues. Thematic synthesis of interviews highlighted version management of data, models, and other machine learning artefacts as particularly critical. The gap in research knowledge and established practices in this area is notable, with an absence of best practices exacerbating difficulties in operationalising machine learning projects. Industry collaborators are exploring and addressing other significant challenges, such as testing and scaling. Additionally, challenges related to organisational collaboration fall outside the scope of this study, which focuses on technical solutions. Nevertheless, the results of the other significant challenges can be helpful for future research aimed at enhancing decision-making and designing robust, integrated data management systems.

Case Study Scenario

A generalised scenario, based on interview results, illustrates the operational challenges of data pipelines and version management within machine learning projects, enhancing understanding: Scenario Description: A small team of 2 - 4 data scientists is tasked with developing a machine learning model, adhering to a continuous improvement process. Initially, the team experiments with static data dumps and local model training. As the project advances, they transition to cloud infrastructure, benefiting from enhanced computational power and increased storage capacity. Throughout this development, the team maintains versioned source code in a shared repository and utilises Jupyter Notebooks for data modelling and Python scripts for data processing. It is important to note that standardised practices have yet to be established for these experimental processes, and the best methods remain under exploration.

General Problems in this Scenario:

1. **Data Versioning:** The absence of a system to track changes to data alongside code complicates tracking modifications over time.
2. **Reproducibility:** The challenge in reproducing experiments arises from poor connectivity between code, data, and experimental details, hindering replicating and validating results.
3. **Collaboration:** The lack of proper tools and processes for sharing experiments, data versions, and their links to code and experimental details can lead to confusion and misalignment within the team, resulting in delays and inefficiencies.
4. **Tool Selection Overload:** The overwhelming number of tools and trends makes it difficult for a small team to select the optimal tools for their needs.
5. **Automation:** Structured workflows are necessary for managing data and various stages of ML development to ensure efficiency, accuracy, and maintainability.
6. **Streamlining Workflow Between Environments:** The team requires a streamlined process for effectively using data and models both on local machines and in the cloud to prevent workflow slowdowns.

7. **Scalability:** Managing large datasets presents challenges, demanding significant data processing capacity that a small team of data scientists may find daunting without better engineering support.

4.4.2 Scope and Goals

This study is driven by the need to address particularly inefficient areas where incorporating MLOps principles can significantly enhance data management and support automation. The primary focus is data versioning, experiment tracking, reproducibility, and collaboration within machine learning projects. These areas have been identified as significant gaps in industry collaboration, presenting opportunities to apply and explore MLOps principles and tools to overcome data and model management challenges, particularly during the model development stage. As noted in the problem identification section, data versioning is intricately linked with other elements, such as experiment tracking and collaboration. It is essential to develop solutions that focus on these interconnected aspects. This research will concentrate on solutions agnostic to the type of machine learning model employed, emphasising a methodology applicable across various model types. This approach ensures that the proposed solutions are flexible and not confined to specific model architectures or data processing techniques. Primary data handling will be conducted through Python scripts, focusing on static data types such as Comma-Separated Values (CSV) files, commonly used in practice. For practical demonstration and to deepen our understanding, a CSV dataset from Kaggle[1] was chosen for tool evaluation. This selection enables consistent insights relevant to similar formats of datasets. However, a detailed examination of the dataset or the complexities of machine learning model configurations is beyond this study's scope. The case study scenario outlined in Problem Identification will be used to identify the conditions and scope of the tool selection process and user scenario demonstration. The overarching goal, derived from stakeholder interviews, is to develop a solution that seamlessly integrates into the prototype model development phase without significant overhead. This solution should ensure smooth integration with existing environments, avoid vendor lock-in, scale efficiently, provide robust support, and remain cost-effective. By achieving this goal, we aim to provide a practical and efficient solution that meets the needs of stakeholders.

4.4.3 Requirements and Acceptance Criteria

To address the identified challenges effectively, specific requirements and their acceptance criteria for solutions are defined based on insights gathered from interviews and iterative feedback from industry stakeholders. These requirements will guide the development of features necessary to address the core problems, and acceptance criteria are formulated to evaluate the effectiveness and applicability of these solutions, ensuring the proposed solutions meet the practical needs of users in real-world scenarios.

[1] www.kaggle.com

Criteria ID	Criteria	Measure
1.1	Tracking Data Version	Demonstration of tool's data tracking and versioning.
1.2	Move/Rollback to Different Data Versions	Rollback scenario demonstration to assess version integrity.
1.3	Team Collaboration on Synchronized Data Versions	Team scenario assessment.
1.4	Open Source	Check tool's open-source license status.
1.5	Cost Effectiveness	Perform a cost-benefit analysis.
1.6	Ease of Use	Collect user feedback from data scientists.
1.7	Documentation and Support	Review GitHub stars, update frequency, user feedback evaluation.
1.8	Low Setup Overhead	Time and resources for setup assessment.
1.9	Vendor Agnostic	Check tool compatibility across different platforms and vendors.

Table 4.2: Acceptance Criteria for Requirement-1

Requirement 1: Data Versioning

Data scientists require a data versioning system that parallels code versioning. This system should facilitate tracking changes in data, maintaining data lineage, and seamlessly transitioning between different versions of data as needed. Additionally, the system should support effective collaboration among team members by allowing them to share data versions. To meet this requirement, we defined acceptance criteria which are detailed in Table 4.2.

Requirement 2: Reproducibility

Data scientists must be able to track and link versions of data, code, and parameters used in experiments comprehensively. They must also be able to reproduce the experiment environments reliably, maintain detailed logs of all experiments for review, and compare them to determine the best reproduction setups. Additionally, they should be able to share these experiments, along with all associated information (data, code, parameters), with their teammates, ensuring that any team member can reproduce these experiments consistently. To meet this requirement, we defined acceptance criteria which are detailed in Table 4.3

Criteria ID	Criterion	Measure
2.1	Comprehensive Experiment Logging	Demonstrate logging capabilities in user scenarios.
2.2	Experiment Reproducibility	Assess reproducibility through re-execution of experiments.
2.3	Team Reproducibility	Test reproducibility across different team members' setups.
2.4	Sharing and Collaboration	Evaluate sharing features and collaboration ease in user scenarios.
2.5	Searchable and Filterable Logging History	Show feature usability by filtering and searching logs in a demo.
2.6	Support for Advanced Features	Evaluate parameter tuning integration in a setup.
2.7	Ease of Use	Collect user feedback from data scientists.
2.8	Open Source	Check tool's open-source license status.
2.9	Cost Effectiveness	Perform a cost-benefit analysis.
2.10	Documentation and Support	Review GitHub stars, update frequency, and documentation feedback.
2.11	Low Setup Overhead	Assess installation time and resource requirements.
2.12	Vendor Agnostic	Check tool compatibility across different platforms and vendors.

Table 4.3: Acceptance criteria for Requirement-2

5 Results

This chapter presents the findings from the Systematic Literature Review (SLR) and interviews and focuses on analysing these sources that highlight the identified challenges. It also explores solutions that address Requirements 1 and 2, as outlined in the Design Science Approach section. The discussion details the tools selected to meet these requirements and demonstrates their effective usage through workflows tailored to specific user scenarios. These scenarios not only validate the acceptance criteria but also emphasise practical application. Additionally, the chapter provides insights and recommendations for each scenario.

5.1 Systematic Literature Review

The initial keyword search in digital libraries yielded 202 results in the Association for Computing Machinery (ACM) Digital Library[1] and 225 in Institute of Electrical and Electronics Engineers (IEEE) Xplore[2]. After removing duplicates, 182 and 206 papers remained, respectively. Subsequently, 8 and 10 papers were selected from each library after applying the study selection criteria. Additionally, five and three literature reviews were identified from respective digital libraries and chosen for inclusion in the snowballing search to find relevant papers.

These 26 papers served as primary sources for the snowballing search, subsequently providing 21 papers from backward snowballing and 4 papers from forward snowballing. The lower number of studies identified in forward snowballing is understandable, as the primary studies were published from 2020 onwards, resulting in fewer citations than older publications.

In total, 43 papers were chosen for thematic synthesis, referenced by identifiers X1 - X43 in this study. These papers and their corresponding identifiers are detailed in the Appendix A.

5.2 Interview

Interviews were conducted with 15 professionals, discussing 13 different systems. An overview of these systems, including their technology domain, developmental state, team size, and data volume, is presented in 5.1. Each system is categorised into one of three developmental states based on the level of functionality available to end-users: ‘Prototype’, ‘Intermediate’, and ‘Production’. In the ‘Prototype’ stage, a minimum viable product is being developed and refined without being available to customers. The ‘Intermediate’ stage signifies that only certain final product features are available for customer use, with ongoing development to complete additional features. Finally, the

[1] https://dl.acm.org
[2] https://ieeexplore.ieee.org/

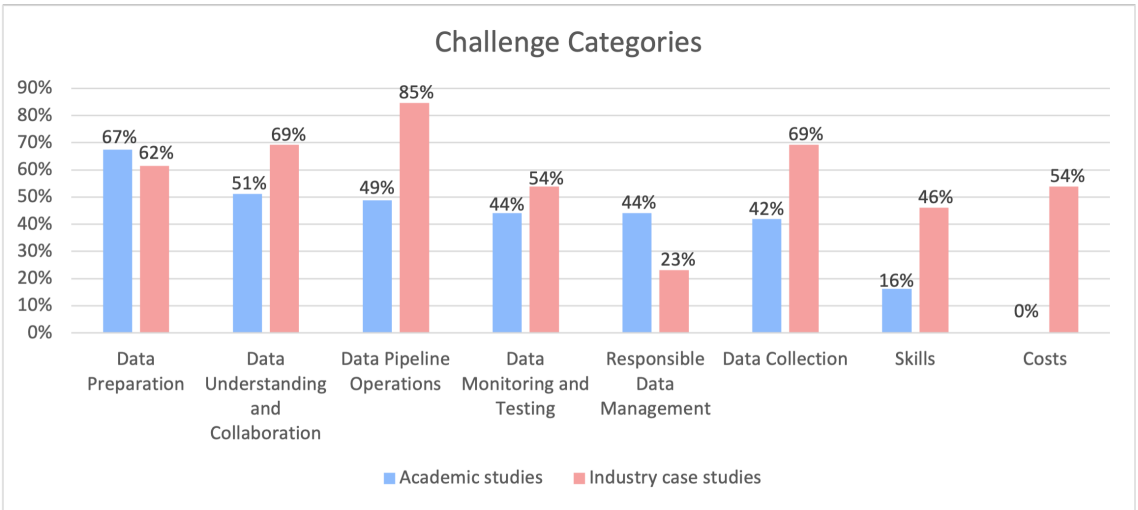

Figure 5.1: Comparison of Challenge Categories Identified in Academic Studies versus Industry Case Studies

'Production' stage indicates that nearly all intended features are available to customers, shifting the focus towards enhancing and maintaining the product. Detailed information on each system can be found in Appendix C.

The practitioners' roles and respective projects are elaborated in Table 5.2. In this research, practitioners are designated by identifiers P1 - P15 and project systems by S1 - S13. The practitioners are employed by a multi-organisational company and belong to various organisations. However, this study does not elaborate on the specific details of these organisations and their relationships.

5.3 Challenges

In our analysis of data management challenges within MLOps, thematic synthesis identified seven major challenge categories from papers selected in the Systematic Literature Review. These categories include 26 high-level challenges, referred to as challenge labels. During a thematic synthesis of interviews, an additional challenge category and two challenge labels were validated, with only one challenge label not observed. The consistency between scholarly focus and real-world challenges highlights significant alignment. Figure 5.1 illustrates these challenges, comparing the frequency of their occurrence in both the academic papers and the case study systems, showing variations that suggest divergent priorities. These variations will be thoroughly examined in the Discussion chapter.

This section delves into each challenge category, providing a detailed examination of the identified challenge labels using insights from the literature and perspectives from professionals gathered through interviews. For ease of reference, the challenge categories are presented in the same order as in Table 5.3. This figure summarises the challenge categories and their labels and references the academic studies and case study systems that underscored these challenges.

Project ID	Category	Project State	Team Size	Data Volume Estimate
S1	Data Engineering and Business Intelligence	Intermediate	2-5	5-10 GB/week
S2	Data Engineering and Marketing Technology	Production	2-5	4-5 TB/week
S3	Data Engineering and Digital Marketing Analytics	Production	10-15	1 GB/week
S4	Data Engineering, Business Intelligence, and Data Governance	Production	10-15	500 GB/week
S5	Data Engineering	Production	2-5	1-10 MB/week
S6	Data Engineering and AI-Driven Product Development	Prototype	2	50-100 GB/week
S7	Data Engineering and Machine Learning	Production	2-5	50-100 GB/week
S8	Machine Learning	Prototype	2	2.6 TB
S9	Data engineering and Machine Learning	Production	2-5	5-10 TB/week
S10	Data engineering and Machine Learning	Production	2-5	5 GB/week
S11	Data Engineering and AI-Driven Product Development	Prototype	2-5	20 MB/week
S12	Data Engineering and Business Intelligence	Production	2-5	1-5 GB/week
S13	Data Engineering and AI-Driven Product Development	Intermediate	5-10	200-500 GB/week

Table 5.1: Case Study Systems Overview

Interviewee code	Project code	Position	Years of Experience
P1	S1	Data Analyst	5
P2	S1	Data Analyst	2
P3	S2	Product Owner	6.5
P4	S4	Product Owner	8
P5	S4	Product Owner	18
P6	S4, S3	Data Infrastructure Engineer	7
P7	S6, S7	Software/Data Engineer	10
P8	S8	Data Scientist	2
P9	S9	Data Scientist	6
P10	S10, S11	Data Scientist	5
P11	S12	Data Engineer	13
P12	S4	Data Architect	14
P13	S13, S10, S4	Product Owner	15
P14	S2, S5, S4	Devops Engineer	17
P15	S4, S2	Executive Manager	21

Table 5.2: Overview of Interviewees

No	Challenges	Academic studies references	Industry case studies systems' references
C1	**Data Preparation**		
	C1.1 Data Processing	X1, X2, X4, X5, X7, X9, X11, X12, X15, X16, X24, X25, X26, X29, X30	S9
	C1.2 Huge Volume of Data	X4, X11, X13, X14, X16, X19, X22, X23, X24, X25, X32, X43	S2 (P3), S4 (P5), S7, S8, S9

Table 5.3 continued from previous page

No	Challenges		Academic studies references	Industry case studies systems' references
	C1.3	Data Labelling	X1, X7, X9, X16, X17, X18, X20, X32	S7, S9, S10 (P13)
	C1.4	Data Cleaning	X1, X2, X16, X19, X24, X32, X36	S2 (P3), S4 (P5), S13
	C1.5	Heterogenous Formats of Data	X1, X8, X11, X12, X21, X25	S4 (P5), S9, S12, S13
	C1.6	Feature Engineering	X18, X26, X27	S9
C2	**Data Understanding and Collaboration**			
	C2.1	Poor Commmunication and collaboration	X2, X4, X11, X22, X23, X28, X30, X34, X36, X37, X38	S1 (P1), S2 (P3), S4 (P5, P14, P15), S5, S7, S9, S12, S13
	C2.2	Data Understanding	X2, X4, X5, X10, X11, X18, X19, X27, X30, X36	S1, S4 (P12), S9
	C2.3	Data Access and Governance	X2, X4, X6, X11, X15, X21, X25, X26, X28, X29, X34	S2 (P13), S4 (P12, P15, P5), S7, S8 (P8), S13
C3	**Data Pipeline Operations**			
	C3.1	Scaling Resources	X16, X19, X21, X22, X23, X24, X25, X30, X31, X33, X34, X35	S4 (P12, P4, P6), S7, S8 (P8), S9, S10 (P10)
	C3.2	Pipeline Design and Integration	X10, X12, X20, X21, X22, X23, X24, X26, X34	S1 (P1), S2 (P3, P15), S3, S4 (P12, P15, P4, P5), S5, S7, S11, S12
	C3.3	Version Management	X13, X22, X23, X30, X33, X34, X36, X37, X43	S8 (P8), S9, S10, S11

Table 5.3 continued from previous page

No	Challenges		Academic studies references	Industry case studies systems' references
	C3.4	Maintenance	X5, X10, X13, X19, X24, X26	S2 (P14), S4 (P5, P12, P14), S9, S10 (P10), S12
	C3.5	Engineering Support	X19, X34	S8, S9, S10 (P10)
	C3.6	Vendor Lock-in		S2 (P14), S5, S8 (P8), S9, S10 (P10)
C4	**Data Monitoring and Testing**			
	C4.1	Data Testing	X1, X5, X8, X9, X11, X12, X15, X16, X23, X26, X27, X28, X29, X30, X34, X40	S1 (P1), S2 (P3, P15), S3, S4 (P4, P12), S7, S9, S10
	C4.2	Data Monitoring and Detection	X7, X9, X11, X12, X16, X23, X26, X28, X34, X35, X38	S1 (P1), S2 (P3), S3, S4 (P4, P12), S7, S10 (P13)
C5	**Responsible Data Management**			
	C5.1	Data Privacy and Personal Data	X5, X6, X10, X14, X18, X19, X21, X24, X29, X25, X32, X33, X39	S2 (P14), S4 (P4)
	C5.2	Data Ethics	X1, X5, X9, X23, X28, X39, X42	
	C5.3	Data Security	X1, X7, X9, X24, X25, X32	S4 (P12), S9
C6	**Data Collection**			
	C6.1	Insufficient Data	X1, X2, X3, X4, X9, X11, X14, X18, X24, X26, X27, X31, X39, X41, X43	S1, S4 (P4), S7, S8(S8), S9, S12

Table 5.3 continued from previous page

No	Challenges		Academic studies references	Industry case studies systems' references
	C6.2	Heterogenous Sources of data	X8, X12, X24	S2 (P3), S7, S9
	C6.3	Data Source Quality or inconsistencies	X24, X26	S1, S3, S4 (P4, P12), S9, S10 (P13), S12
	C6.4	Data Requirements	X3	S1 (P1), S2 (P3), S4 (P4, P15)
C7	**Skills**			
	C7.1	Skills Competency	X4, X6, X10, X11, X13, X22, X43	S1, S2 (P14), S4 (P4, P12), S7, S9, S11
C8	**Costs**			
	C8.1	High Costs		S1, S3, S4 (P4, P5, P6, P12, P13), S6, S8 (P8), S9, S10 (P10)

Table 5.3: Data management challenges referenced in academics and industry

C1 Data Preparation

This category covers a variety of challenges related to cleaning, processing, feature engineering, and data labelling. It also considers the effects of data attributes such as type, format, and size, which significantly impact these activities and contribute to the complexities and inconsistencies encountered.

C1.1 Data Processing

Academic Insight: Paper X16 described the problem that most practitioners in the industry avoid using specific ML data analysis tools due to the lack of universally applicable solutions that can meet domain-specific and diverse data processing requirements. Paper X11 brings to light the issue of incorporating incomplete knowledge or false assumptions in data processing, leading to a misrepresentation of the real world and introducing biases into ML models. Papers X5 and X7 discuss the biases that can arise, mainly from using data isolated in silos. Paper X2 reported that a common issue identified by practitioners is the training-serving skew, where models perform well on the given training data but struggle with production data, stemming from insufficient training

data and unclear representation of the production environment. Furthermore, paper X9 emphasises the problems associated with preprocessing data with imbalanced features, highlighting the need for sophisticated methods to rectify these imbalances and ensure model accuracy. Papers X4, X26, X30 noted the challenge of extensive processing required to manage data quality from external sources. At the same time, papers X12 and X15 note the multi-dimensional nature of poor-quality data, complicating the comprehension and consistent processing of it. Paper X1 emphasises that, despite thorough processing, data quality problems can still remain due to inherent uncertainties. Paper X29 raises concerns about maintaining uniformity of preprocessing across different environments, emphasising the need for alignment between development and production settings to fulfil the model's requirements.

Industry Insight: Data Scientist P9 also discussed the difficulties in data normalisation across multiple tables, which often requires iterative refinement to achieve a suitable data format for analysis. The challenge of handling data bias, particularly with extreme class imbalance, was mentioned: "We have this very extreme class imbalance with people who convert versus people who do not convert. That is definitely always an issue," explained P9 about project S9, underlining the complexity in managing disproportionate data supporting the findings in academics.

C1.2 Handling Large or real-time data

Academic Insight: Managing large volumes of data presents its own set of challenges. Paper X19 talks about the complexities of cleaning extensive datasets, where involving domain experts can inadvertently introduce bias. Papers X16 and X32 discuss the logistical difficulties of labelling large datasets, a process often overwhelmed by the sheer volume of data. Papers X13 and X22 highlight the challenges in efficiently processing and storing vast amounts of data, advocating for implementing automated systems to aid in continuous data preparation and facilitate model retraining. Paper X24 touches on the real-time data preparation dilemma, balancing the need for timely processing with maintaining high performance and speed. Papers X25, X23, and X4 described that in real-time data systems, maintaining data quality is challenging due to the dynamic nature of sources, leading to inconsistencies and errors that complicate data processing and its use. Papers X11 and X43 describe the struggle with managing real-time data, which can be messy and variable, affecting overall system performance. This challenge is exacerbated in large-scale environments, as noted by X1, X14 and X43. Additionally, paper X43 also points out the need for high-quality hardware and software to manage the preparation of large-scale real-time data that is often hampered by budget constraints and the trade-offs with system complexity.

Industry Insight: The volume of data represents a considerable challenge, affecting various aspects of data preparation. Practitioner P7 about system S7 comments on the complexity due to size: "So the size of data makes it quite complex to make sure everything is as expected." P7 and P3 mentioned that preparing and cleaning huge volumes of data according to the requirements is problematic for systems S7 and S2, respectively. This problem was further supported by data scientist P8, who mentioned,"This is a challenge [to] deal with this massive data for training" when discussing challenges in system S8. Furthermore, a large volume of data poses performance challenges (P5, S4).

Also, consistent with the narratives in academic research, P8 about system S8 discusses the real-time data processing issues, "[...] if I give you a session ID, you need to give me all of the events for that session ID and then probably this is annoying in real time, because if I do this [. . .] sometimes it just takes [some time and becomes a bottleneck]". The urgency of real-time data processing, especially during inference, remains a prominent challenge. Practitioner P7 raises a similar concern regarding the need for near real-time data processing capabilities in pipelines that provide the most up-to-date information without compromising the cost and system performance of system S7.

C1.3 Data Labelling

Academic Insight: Data labelling is a crucial but challenging aspect of data preparation. Paper X16 identifies a significant concern regarding the requirement for standardised guidelines in labelling, which threatens data quality. This issue is exacerbated by the lack of robust validation mechanisms, as emphasised by papers X16, X17, and X32, underscoring the necessity for reliable and consistent data labelling practices in the industry. Papers X9 and X7 discuss the technical challenges in detecting, preventing, and correcting inconsistencies in labelled data. Paper X16 also raises concerns about the significant cost and time required for data labelling, worsened by the absence of automation.

Papers X20 and X32 highlight ongoing challenges in manual labelling, particularly for tasks needing frequent model retraining, such as short text classification and extreme multi-label text classification. The need for domain expertise in labelling is critical, yet finding experts willing to undertake this task is increasingly difficult, as noted in paper X17. Paper X18 criticises the support of less qualified individuals, like interns, for labelling, risking the data's trustworthiness and quality. On the same note, papers X1 and X16 point out the adverse effects of noisy or missing labels, underlining the urgent need for comprehensive strategies to address these issues and mitigate potential biases. Paper X17 emphasises the difficulty in predicting evolving data labelling needs and applying datasets across diverse machine learning systems, underscoring the complexity of ensuring data remains relevant and well-prepared.

Interview Insight: The labour-intensive nature of data labelling is recognised as a significant challenge. P7 about system S7 describes the process as "[...] very challenging because it is very time-consuming, and it is not a glorious task [...] so it is very hard to convince our colleagues to help us," highlighting the substantial effort and persuasion needed to undertake this critical task.

Another emerging challenge is the need for consistent numerical encoding of categorical data throughout training sessions and inference phases to facilitate accurate tracking and comparison (P9, S9). In the realm of AI chatbots, product owner P13 discusses the ongoing challenge of integrating human feedback into the annotation process to enhance the performance of system S10.

C1.4 Data Cleaning

Academic Insight: Data cleaning is labour-intensive and time-consuming, highlighted by X2, X36, and X24, who stress the significant investment required to address poor data quality. Paper X16 echoes this sentiment, adding insights into the challenges of standardising and automating data cleaning tasks, which are essential for scalability and efficiency. Furthermore, paper X32 discusses the difficulty of eliminating poisoned or sensitive data, underscoring the need for effective

methodologies to ensure data purity and integrity. Paper X16 also points to the challenges in developing tools adaptable to diverse data types, which is crucial for thorough and domain-specific cleaning processes.

Industry Insight: Data cleaning involves not only the removal of irrelevant information but also the strategic selection of valuable data. Practitioner P13's discussion of system 13 highlighted the complexity of data cleaning, which involves manually setting filters for various sources to segregate and preserve valuable content while discarding irrelevant or unnecessary details.

The task becomes particularly challenging when dealing with datasets that have a large number of fields. Determining and selecting the essential fields from a vast range of fields is difficult (P3, S2). These insights are in line with academic research on the subject.

Practitioner P3 about system S2 elaborates on this challenge: "We also need to think about which data we actually need from that new system because most of the time they have a whole bunch of different data fields, a lot of which we do not actually need and do not want to use in the future. We always try to discard them right at the start; otherwise, we end up with huge databases filled with about 80% unusable or unnecessary data."

C1.5 Heterogeneous Data Formats

Academic Insight: The diversity of data formats poses significant challenges in maintaining high-quality data, as highlighted by papers X25, X8, and X12. These papers emphasise the complexities of managing large datasets with varied data types and formats and stress the importance of setting consistent standards for size, shape, format, and data type to preserve data integrity. Similarly, paper X21 raises concerns about establishing a common data format, which echoes the difficulties in achieving standardisation across diverse datasets. Additionally, paper X1 notes the complexities of handling multi-modal and dynamic data.

Interview insights: The issue of heterogeneous data formats is a recurring theme in interviews similar to academics, underscoring the critical need for seamless integration and uniformity across diverse data sources. Practitioner P13 highlights the lack of a standard format as a challenge in system S13: "[...] there is no common format, so that is also a challenge to agree on a common data exchange format." This sentiment is echoed by P5 about system S4, which faces similar difficulties.

The main challenge involves integrating various data formats to ensure consistent and reliable analytics despite differences in granularities, notations like currencies, and time zones, as evidenced by the experiences of practitioners P9, P11, P13 about systems S9, S12, and S13, respectively. P11 about system S12 elaborates on the issue, emphasising the complexity of unifying data from different countries and time frames: "Then you have challenges that from different countries, different time frames, whatever you have to unify this thing [...]."

Furthermore, P9 about system S9 mentions the lengthy and intricate process of preparing data marts in the correct format, highlighting the significant effort required. Another aspect of the challenge is the need to adapt to new data formats resulting from changes in source systems, which can make previous data sets obsolete, particularly when these updates involve the removal of dimensions, as discussed by P11 about system S12.

C1.6 Feature Engineering

Academic Insight: In the realm of feature engineering, the challenges are multi-dimensional. Papers X18, X26, and X27 discuss the intricacies of selecting the right features, noting the dual challenges of the high costs associated with integrating new features and the potential for performance degradation when inappropriate features are chosen. This selection process is critical, as it requires a deep understanding of the data's underlying structure and the information it conveys.

Industry Insight: Practitioners, like academics, find feature engineering complex, noting discrepancies between expectations and reality. For instance, data scientist P9, regarding project S9, highlighted, "It's this obvious feature engineering that you need to do. It's not exactly how you expect it to be," underscoring the unexpected intricacies involved.

C2 Data Understanding and Collaboration

This category discusses the challenges associated with understanding data from diverse sources and facilitating effective collaboration among teams and stakeholders. It includes issues related to data access, governance, and collaborative practices. Additionally, this category addresses the complexities of communication, cultural, and organisational practices that impact the interpretation of data and decision-making processes related to data systems.

C2.1 Poor communication and collaboration

Academic Insight: Papers X11, X30, and X36 emphasise the importance of interdisciplinary expertise and the communication difficulties among stakeholders from different organisations and domains. Papers X2 and X28 note significant challenges with disagreements on data quality and quantity. X2 described that insufficient documentation can lead to misaligned expectations between model and data teams. Paper X28 adds that stakeholders' reluctance to document data requirements worsens these issues. Meanwhile, paper X4 points out that unclear data hand-off processes can create complex dependencies and misaligned goals among stakeholders, further complicating collaboration.

Additionally, papers X28 and X11 noted that data teams are often interested in collecting data without understanding requirements. This often forces them to map the requirements to the already collected dataset. Moreover, papers X22 and X38 discuss the problems caused by organisational silos and stress the need for enhanced collaboration to clarify and streamline communication. Similarly, paper X23 advocates for improved teamwork across different groups to integrate efforts in designing and developing machine learning system pipelines, ensuring a comprehensive data and feature engineering approach. Lastly, paper X37 highlights the challenges of multiple collaborators working on the same Jupyter Notebook, underlining the necessity for efficient version control to facilitate smoother collaboration.

Industry Insight: Communication and collaboration issues often lead to duplicated efforts in data preparation, as observed by P5 about system S4 and P3 about system S2. A comprehensive data catalogue to mitigate these issues faces obstacles due to distributed data and stakeholder engagement,

as practitioner P3 about system S2 explains: “We are building a data catalogue [...] But it is super hard to get everyone on board and get a big overview of this because it is [very] distributed across the whole company.”

Poor communication methods, such as email exchanges between multiple organisational entities, can cause misunderstandings and assumptions, expanding the communication gap. Practitioner P14 about system S5 highlighted this issue, indicating that ineffective communication modes could obstruct data-related interactions and decision-making processes. Moreover, the miscommunication between different teams, particularly between data producers and consumers, creates a significant gap in understanding and leveraging data effectively. P7 about system S7 notes, “There is a bit of a disconnect [...] it is a bit hard for ourselves, colleagues or people working on the product to understand the benefits of using data.” Communication with the business side and technical side teams is discussed as challenging due to differing expectations, perspectives, and requirements. While the tasks may appear straightforward, achieving the objectives is often complicated by technical complexities, as mentioned by (P1, S1)(P3, S2)(P11, S12)(P14, S4)(P15, S4). However, timely communication has been identified as a mitigating factor. As the example mentioned by P14 about system S5 regarding inadequate communication and understanding, [which leads to inefficiencies, such as] unoptimised Online Analytical Processing (OLAP)) systems with long query times, Similarly, Coordinating with international and multi-organisational stakeholders necessitates extensive communication, especially when navigating access permissions. In this context, concerns about digital rights access were mentioned, adding layers of complexity, as noted by P13 about system S13.

Moreover, tools like the Jupyter Notebook, while beneficial for exploration, limit simultaneous collaboration on data and models, posing an ongoing challenge, as mentioned by P9 about system S9.

C2.2 Data Understanding

In this challenge, the primary concerns revolve around documentation quality, metadata availability, and domain expert communication, significantly impacting data reliability and model trustworthiness.

Academic Insight: X4 and X30 have raised issues about poor documentation quality, noting that unclear definitions undermine trust and contribute to the risk of developing unreliable models. Similarly, papersX11 and X10 have observed that inadequate documentation and missing information compromise data quality, especially when data is gathered from diverse sources.

Papers X11 and X5 highlight that when practitioners are forced to make assumptions due to reasons such as the lack of comprehensive metadata, it can compromise data and model quality and potentially lead to biased outcomes.

Moreover, the availability of domain experts is a critical bottleneck in understanding data, as discussed by papers X4 and X18. Paper X27 elaborates on how data scientists often spend excessive time hypothesising due to insufficient communication with these experts. This communication gap not only delays the data analysis process but also hinders the development of accurate and effective data models.

Industry Insight: The understanding of data, which is critical to creating accurate models, involves complex decision-making and thorough data mapping to the problem context. Data Analyst P1 about system S1 highlights the significance of the exploration phase, emphasising that neglecting this step can lead to costly repercussions later on. Moreover, knowledge transfer often occurs through informal channels, leading to gaps in understanding and errors in data modelling, a situation mentioned by both P1 about system S1 and P9 about system S9.

Data understanding can be difficult, particularly when working with data prepared by others with limited documentation, making it hard to interpret and use effectively. Data Analyst P2 about system S1 shares this frustration: "I think the biggest challenge I had was to understand which data is used [...] we are using quite a bit of different data sets here, and there was not a proper documentation for that."

Furthermore, the lack of standardised processes and systems hinders the process of finding and utilising data. This is exemplified by data scientist P9 about system S9, who discusses the difficulties caused by non-standardised naming conventions and the absence of a structured data documentation approach, leading to inefficiencies and obstacles in accessing and understanding data needed for analytical purposes.

Moreover, the absence of a centralised and user-friendly system for data documentation presents a significant barrier to effective data understanding. Data architect P12 elaborates on the complexities of initiatives to implement a data catalogue in system S4, noting the organisational struggles with maintaining a coherent and accessible database: "We try to introduce so-called data catalogues [...] However, we just realised this is super complex to implement because we have a lot of data sources and a lot of data [. . .] So the tables and the data are not really organised, so that it's basically not usable."

C2.3 Data Access and Governance

Academic Insight: Challenges in data access and governance within organisations often stem from misaligned business objectives and a lack of awareness, with X4 noting the difficulties in facilitating data sharing between teams. Paper X2 adds that privacy and security concerns further hinder data access, with these challenges being partly attributable to past collaboration difficulties and a prevailing lack of trust among stakeholders. Paper X6 highlights how data quality issues lead to sharing resistance, suggesting that standard guidelines and agreements are needed to build trust. In parallel, X34 emphasises the necessity for improved collaboration and knowledge sharing among team members to facilitate effective data management. Papers X21, X6 explores the complexities in multi-organisational data collaboration, which are affected by differing privacy laws and cultural practices.

The need for a robust data governance strategy is underscored by paper X25, who notes its importance for data-driven systems despite the ongoing challenges in its implementation.

Furthermore, paper X4 delves into the complications that arise from unclear responsibilities within centralised data structures, leading to poor data maintenance and quality and ultimately preventing full data utilisation. Paper X15 highlights the issue of absent data quality ownership, particularly in large or outsourced data environments Lastly, paper X28 addresses the necessity of data maintenance, acknowledging the financial constraints that often restrain these efforts.

Industry Insight: Finding and accessing the right data sources within a large, distributed organisation is challenging, as stated by product owner P3 about system S2: “And one big issue is first of all, getting the information like getting access to the systems and sometimes quite hard because we’re such a distributed huge company, it’s hard to find who’s responsible for it [...]” This experience is also supported by practitioner P8 about system S8 who faced the same challenge.

The challenge involves establishing a balance between data accessibility and control, as increasing user numbers and data volume with unrestricted self-service led to unmanaged costs and a shift to a more managed approach with clear guidelines, resource monitoring, and governance can ensure efficient and responsible data usage(P5, S4)(P12, S4). Data architect P12 about system S4 emphasises the granularity of access control: “The only problem is right now, that the roles are not perfect [. . .] how fine granular do you want to get the access to the data sources? [. . .]”

Furthermore, lack of access and control leads to operational inefficiencies. P7 about system S7 explains the consequences of such limitations: “[...] because often basically we will need something or we will observe an issue in the data and often this will take a lot of time on the other side, to get confirmed and fixed.”

As practitioners P12 and P15 discussed, managing and maintaining numerous integrated data sources where clear ownership is lacking makes it difficult to ensure data quality and format consistency in system S4. Executive manager P15 articulates this about system S4: “Responsibilities like who is responsible for which type of data and who makes sure that what type of data or did the contract is fulfilled, and that is the common issue that I have seen everywhere.”

C3 Data Pipeline Operations

This category explores the challenges of designing, building, and managing data pipelines and operations. These challenges encompass designing and integrating versatile pipelines, managing scalability as data volumes grow, ensuring consistent version management, providing necessary engineering support, and maintaining system reliability and efficiency.

C3.1 Scaling

Academic Insight: X25, X19, X23, and X30 highlight scalability challenges, primarily with real-time and continuously growing data. Paper X22 urges the need for auto-scaling infrastructure to manage this expansion. Paper X31 identifies scalability challenges in reinforcement learning systems due to distributed architectures and high-dimensional data. Papers X35 and X33 discuss the need for a resource-efficient approach to developing scalable platforms and maintaining AI operations in response to continuously increasing data. Furthermore, paper X24 highlights the critical need for performance-optimised data pipelines, primarily when time sensitivity is crucial. Additionally, X24 emphasises the importance of testing data pipelines for scalability, noting the complexities introduced by distributed architectures.

Paper X16 reflects that available tools are not often sufficient for the scalability requirements of continuously growing data. Paper X34 further discusses complexities due to tool dependency and lack of modularity.

Paper X34 addresses the significant scaling challenges encountered when transitioning from development to production, advocating a shift from a model-centric to a pipeline-driven approach.

Industry Insight: The scalability of pipelines, particularly in handling unexpected traffic spikes, is problematic. P4 and P6 about system S4 pointed out the limitations in auto-scaling capabilities, with P6 detailing the challenges with Kinesis: "[...] Kinesis as we're using it right now, is very rigid, we always have to sort of like anticipate how much traffic will be there at any point in time and configure the shards inside Kinesis. [...] We must always scale those in advance so they are not autoscaling." Data scientist P10 about system S10 highlighted the inefficiencies in a batch processing system, where complete reprocessing is required for any single error, leading to resource wastage and underscoring the need for a system redesign to improve scalability. The scalability challenge was further compounded in batch processing systems, as described by data scientist P10 about system S10. The requirement for complete reprocessing in case of a single error leads to significant resource inefficiency and necessitates a system redesign to improve scalability.

For real-time streaming pipelines, the necessity of scaling according to data size while maintaining performance and cost-effectiveness(P7, S7)(P8,S8). Data scientist P8 about system S8 underscored this issue, stating, "First challenge is a lot of data." This comment reflects the daunting task of managing large data volumes in scalable systems.

Further complicating the landscape, P12 about system S4 discussed the scalability limitations of the workflow orchestration tool, Airflow, which was not adequately scalable, leading to system bottlenecks. This necessitated a transition to more robust, containerised solutions capable of handling increased loads and offering greater reliability.

C3.2 Pipeline Design and Integration

Academic Insight: Paper X24 underscores the need for a versatile toolset that supports integration across the diverse and expanding domain of data engineering workflows. The challenge lies in selecting appropriate tools or frameworks for building data pipelines, with considerations varying widely depending on the use case. The absence of a comprehensive framework to manage both online and offline data exacerbates maintenance burdens, necessitating separate pipelines with different computing engines. X12 noted that a rapid increase in tools and libraries may lead to confusion and potential errors rather than facilitating the correct tool selection. The issue is compounded by frequently updated APIs and outdated tool documentation, hindering their effective utilisation. Paper X20 further elaborates on the compatibility issues encountered with different pipelines, underscoring the complexities of developing robust and flexible pipeline systems.

Paper X26 highlights the reliability problems caused by poorly designed transformation stages in data pipelines. Additionally, paper X23 noted uncertainty arising from data variability, which introduces complexities to architectural design that are not present in traditional software systems. Paper X10 also discusses the challenges with the architecture design of ML systems because of fragmented data infrastructure. Moreover, legacy systems add another layer of complexity to this problem. Furthermore, paper X34 emphasises the increase in complexity when integrating machine learning due to factors like the lack of modularity and the extensive dependencies on tools and data. Paper X22underscores the complexity of integrating multiple software and hardware stacks,

underlining the need for automation strategies to streamline these processes. Paper X21 further emphasises the difficulty in managing data operations, particularly when datasets are large and cannot be moved easily between organisations.

Industry Insight: The proliferation of tools in the market complicates the selection process, as noted by P4 about system S4: "So obviously there are different companies also providing those tools and just to take the time and investigate stuff, this is really challenging because there are just so many things and sometimes we forget the actual use case you know [...]." The decision-making process is further complicated by the need to balance diverse opinions and preferences (P15, S4). Setting up new big tools, which require a lot of setup, can also be challenging and time-consuming (P3, S2). Integration issues, especially concerning version compatibility between different tools, were highlighted by P1 about system S1. Additionally, P6 mentioned the obstacles of tools lacking native communication capabilities, necessitating workarounds or additional integration efforts when discussing system S3. Moreover, the inefficient tools that are error-prone and have missing functionalities act as major obstacles(P3, S2)(P14, S5). Another challenge is when a tool is unstable or lacks features, leading to unexpected issues and necessitating a switch to alternative solutions (P11, S12). The issue of tool stability was further underscored by P11, who needed to switch to an alternative solutions system due to experiencing these challenges in system S12. Data infrastructure engineer P6 about system S3 discussed the difficulties associated with using tools that lack maintenance and are burdened with outdated plugins which lack support. Access management complexity is another significant concern in tool integration, with product owner P5 about system S4 detailing the challenges in managing access effectively across various tools. P15 about system S2 underscored the importance of designing pipelines that facilitate data flow tracking and maintain data schema consistency, which is essential for efficient data management and integrity. Furthermore, P5 about system S4 discussed how the growing integration of tools and components complicates architecture design, affecting both the maintenance and future usability of the pipeline solutions.

P7 also highlighted the complexity of managing a pipeline that involves multiple steps and systems, stressing the difficulties in achieving a real-time overview in system S7: "So it's not having a real-time view of the whole pipeline; it is [the biggest challenge]. So there are many steps in the middle, and all of them kind of rely on the [previous step], and if something [changes], we would have troubles in one of these steps."

Practitioners P10 and P12 emphasised the need for continuous updating and standardising practices and tools to facilitate better solution design and collaboration for systems S11 and S4. P10, regarding system S11, reflected on this necessity: "I guess the challenges are to define standards there for yourself, for your team, and then agree on that and use this."

C3.3 Version Management

Academic Insight: Challenges in managing machine learning projects, particularly regarding data model and provenance information, are highlighted by papers X13 and X22. X23 adds to this by emphasising the need for traceability and co-versioning in development pipelines. Version management is pinpointed as a significant challenge in MLOps by papers X37, X43, and X30, detailing the struggles with versioning data, artefacts, pipelines, and collaborative tools like Jupyter notebooks. Furthermore, paper X34 points out a notable gap in tool support for tracking and

managing data provenance effectively. X33 stresses the need for traceability of models, and X36 addresses the challenges faced by version management tools in ensuring consistency across various environments.

Interview Insight: The absence of automation in the development stages of data pipelines presents significant challenges for manual tracking across data collection, training, and deployment. P8 about system S8 discussed the complexities of managing version changes when new data is updated in the source, succinctly noting, "If I change anything in the pipeline at some point, everything else needs to be rerun." The manual process of version management in pipelines is not only challenging but also comes with practical difficulties. As data scientist P9 about system S9 shares, "And usually what we do there is we just have the very old pre-version control setup that you copy a file, and then you work on that newer version and make sure you have this old one or your comment out thing. So that's really suboptimal [. . .] when we do more explorative stuff." This highlights the need for a more efficient and streamlined approach to version management.

The lack of a systematic approach during the exploration phase to version control in the preprocessing and sharing of data files often leads to duplication and confusion over file versions and their creation details (P10, S11)(P10, S10)(P13, S10). It is important to establish a data versioning system to manage multiple iterations of training datasets for continuous model training in the future (P10, S10). As data scientist P10 emphasised, "We [do our experiments] and have different versions of the data, maybe different dataset dumps[...]." The practitioners expressed their desire for a versioning system integrated with Git to better manage code, models, and data in experiments, thus enhancing reproducibility. However, this system has not been implemented yet (P8, S8)(P9, S9).

C3.4 Maintenance

Academic Insight: Paper X26 discusses the complexities of troubleshooting data pipelines, pinpointing the problem to an inadequate grasp of how data transitions between stages. Additionally, X26 addresses the obstacles in managing machine learning systems, especially the challenges posed by the lack of detailed metadata and the fragility of data pipelines, which lead to technical debt and undermine the system's reliability. Paper X13 noted challenges in maintaining and refining machine learning systems as they evolve. Both papers X24 and X10 talk about the increased maintenance efforts required when multiple frameworks or tools are employed in machine learning systems. They propose integrating these frameworks or developing a comprehensive framework as possible remedies, although each comes with its own set of engineering hurdles. Meanwhile, paper X5 discusses how the reliance on glue code, necessary for merging various data validation processes, can inadvertently introduce technical debt and challenges to the engineering workflow.

Industry Insight: Practitioners highlighted the necessity of creating systems that are resilient, fault-tolerant, and furnished with efficient disaster recovery strategies, all while managing costs (P10, S10)(P12, S4). P10 from S10 articulated the complexity involved in contingency planning: "What happens if something goes wrong, and I need to roll back? Keeping all these considerations in mind and having a Plan B is quite challenging from a conceptual perspective."

Debugging proves particularly demanding when a pipeline crash stems from the intricacy of multiple infrastructure layers, as P5 from S4 observed. Resolving such incidents is both complex and time-consuming.

Upgrading pipelines also presents its own set of challenges, notably when component incompatibilities lead to system failures. P9 from S9 noted, "[...] version upgrades can sometimes be a problem, and then you have breaking changes in some [...] AWS package for Python or whatever." In the context of maintaining operational continuity, P14 about systems S2 and S4 detailed the difficulties encountered during upgrades in Kubernetes environments: "When it comes to upgrades, [...] we mostly using the Kubernetes now and [...] this is always a challenge. You find a good way that you have a zero downtime[...]."

Furthermore, the effectiveness of disaster recovery systems is crucial; their failure to be properly set up or maintained can lead to significant data loss in critical situations, a concern shared by (P11, S12).

C3.5 Engineering Support

Academic Insight: X19 emphasises the need for software engineers to aid data scientists in developing data pipelines, given that such development requires software engineering practices beyond the usual expertise of data scientists. X34 echoes this sentiment, highlighting the importance of engineering support to improve data scientists' management of infrastructure and tools.

Industry Insight: Handling large volumes of data in data pipeline operations necessitates the use of Spark or similarly complex technologies, which in turn requires more engineering support for data scientists, as noted by (P9, S9), (P8, S8), and (P10, S8). P9 about S9 explained the challenge: "So working with this type of data is really challenging [...] You always should go through data engineers, and usually you should also use a Spark job because Python would also usually be too slow for that." Data scientist P8 from S8 highlighted the trade-offs made to simplify data handling, which might affect the scalability of the solutions: "The first challenge is it is a lot of data, and so I think the way we did it probably would not scale in a way because what we did is one of the reasons why we use Polars and not something like Spark is just to reduce the complexity of the problem." Additionally, the challenge lies in finding a balance between rapid development and the robustness required for production environments. P10 about system S10 emphasised the dilemma of needing to avoid shortcuts during initial development phases that could later necessitate rework, thereby complicating the project's time and complexity.

C3.6 Vendor Lock-in

This challenge is highlighted only during interviews. It underscores the inflexibility and integration barriers stemming from vendor lock-in, complicating interoperability across products from different vendors. This challenge includes difficulties encountered during interoperation in environments such as multi-cloud and hybrid-cloud systems.

Industry Insight: The challenge encompasses issues associated with vendor-specific constraints, interfacing difficulties in multi-cloud or hybrid-cloud environments, and the general inflexibility posed by vendor dependencies. Data scientists like (P8, S8) and (P9, S9) note the difficulties in maintaining tool flexibility across various local and cloud environments, which often leads to synchronisation issues and increased overhead. P8 about S8 expressed frustration with the operational inefficiency due to these limitations: "And to sync everything, I didn't want to rerun everything again, this was super annoying."

P9 about S9 detailed the complexities encountered when transitioning to multi-cloud environments, especially in terms of setup exploration, permission management, and security: “If we’re doing a more explorative thing on remote machines, like Amazon Web Services (AWS) Elastic Container Service (ECS) instance or something, then it’s a little bit more complicated [...] we have a multi-cloud setup and then permissions and all these things.” These multi-environment complexities also affect version control, adding layers of difficulty in managing pipeline operations.

Data scientist P10 about S10 discussed the challenges related to data transfer latency in hybrid cloud setups, emphasising the need for seamless, vendor-agnostic solutions to mitigate these issues. Vendor lock-in also leads to high costs and reduced flexibility, constraining the choice of tools and technologies (P10, S10)(P14, S2).

Additionally, practitioner DevOps engineer P14 about system S5 pointed out the challenges with closed-source software, which lack community support and thus complicate their use in specialised cases.

C4 Data Monitoring and testing

This category encompasses the complexities involved in processes designed to identify quality issues in data and ensure its integrity, which is critical for optimal model performance. It includes activities related to monitoring data and reporting on its status. The discussion also delves into various anomalies, detailing their impacts on data quality and subsequent model effectiveness.

C4.1 Data Testing

Academic Insight: Paper X26 discussed the complexity and time-consuming nature of evaluating data quality, including labels and formats. Similarly, papers X23 and X3 highlighted industry concerns regarding the methodologies for testing the quality of data, models, and their artefacts. X1 pointed out the concern of no data quality guarantee even after cleaning and validation. Paper X40 claimed difficulty in assessing quality in high-dimensional and discrete data, noting that automated test input generation falls short of providing a cost-effective solution for gathering test data. Papers X5 and X8 underscore the risk of production downtime due to broken pipelines, often due to inadequate data type checks, emphasising the necessity of data type validation. Similarly, paper X15 point out that the use of default settings in data collection programs complicates the process of tracing errors back to their sources. Furthermore, papers X5 and X11 discuss the intricacies of bias detection during preprocessing, necessitating domain expertise and a deeper understanding of the data to accurately reflect real-world scenarios. X28, along with papers X9, X27, X16, and X29, also delve into the difficulty of determining data appropriateness for specific problem domains and measuring the training data’s completeness in relation to the operational environment.

Industry Insight: Anomalies are typically identified only during initial explorations, with a noted lack of any specific tests written for anomaly detection, a concern raised by (P10, S10), (P13, S10), and (P9, S9). Practitioner P12 discussed the main challenges in testing data in system S4, including determining the depth of documentation and testing, the timing and frequency of tests, the lack of automation tools, and the need for tailored approaches for each data source. Statements from

participants like “[...] the size of the data is the main challenge” (P7, S7), “tests [...] needed to be maintained” (P4, S4) and “[...] writing these tests is like super much effort [...]” (P3, S2) reflect on the practical testing efforts and maintenance requirements.

The absence of initial validation tests in data pipelines can increase their fragility and complicate debugging processes (P15, S2).

The desire for automation stems from the limitations of manual tracking, which can miss issues, leading to system downtimes (P4, S4). Automating testing has inherent challenges due to the unpredictability of anomalies, particularly with unreliable external data sources(P3, S2)(P6, S3). Practitioner P6 about system S3 mentions the trust issues in data quality and the unclear means to verify problems, underlining the necessity for intelligent testing or observative solutions. P7 plans to use data observability tools in the future to automate and establish trust within the system S7. Practitioner P1 from S1 also described manual testing as a significant pain point, expressing hope for potential automation leveraging AI’s capabilities.

C4.2 Data Monitoring and Anomaly Detection

Academic Insight: X23 points out the added complexities of data monitoring within ML systems and their components. X34 mentions the team’s lack of support for monitoring and detecting training-serving skew, also noting delays in implementing necessary monitoring measures despite their recognised importance. Additionally, papers X11 and X16 highlighted the systems’ lack of robustness in managing evolving or dynamic data.

The security aspects of data monitoring are discussed by X7 and X9, who explore the challenges in identifying security risks like dataset poisoning, backdoor attacks, and data leakage.

X35 speaks to the ongoing difficulties in detecting and mitigating data leakage and concept drift. Papers X26, X28, and X38 discuss the challenges associated with data drift, such as model performance degradation and emphasise the need for reliable models and data validation strategies. Papers X12 and X34 highlight the limited tool support for continuous monitoring to ensure data quality, highlighting a gap in continuous data quality assessment.

Industry Insight: Silent failures in data pipelines pose monitoring challenges, with alert systems sometimes failing to detect issues (P12, S4). While dashboards and manual monitoring are preferred over complex test writing (P3, S2), the lack of automated sanity checks and alerts necessitates manual interventions (P7, S7), raising concerns about data integrity, especially when bad data from the source doesn’t trigger errors (P6, S3).

The operational difficulty of tracking numerous processes and the complexities of reporting are significant concerns (P3, S2)(P4, S4). As P3 about system S2 described: “[...] difficulty for us is keeping track of everything [...]. I don’t know how many [processes and jobs are] running every day, and the reporting part of it is always also a hassle”.

The necessity of deciding what data aspects to monitor and the challenges of manual tracking are elaborated by P1 about system S1 and P4 about system S4, indicating the overarching need for efficient monitoring systems.

For AI chatbots, product owner P13 about system S10 underscores the challenge of developing automatic methods to detect issues like toxic behaviour in conversation data, given the impracticality of manually reviewing extensive user interactions. The practitioner reflects on the need for automated systems to identify and alert qualitative issues in data: "Is there a way to automate this and get alerts when something strange is happening, not in a technical sense, but in a qualitative sense?"

C5 Responsible Data Management

This category includes the complex challenges of ensuring data privacy, security, and integrity. It encompasses the difficulties of adhering to ethical standards and compliance with legal requirements.

C5.1 Data Privacy and Regulations

Academic Insight: Paper X6 raised concerns about ensuring privacy when utilising public data. Papers X10 and X39 highlighted the intricate challenges posed by the lack of clear legal guidelines for using personal data, which complicates the process for data scientists to discern permissible data usage. Papers X19 and X18 discuss how the necessity of sensitive or confidential information for data understanding can be obstructed by privacy concerns, making data analysis more time-consuming. Paper X5 highlighted the dilemma in understanding the appropriate trade-off between accuracy and privacy.

Paper X14 points to the hurdles in meeting real-time data requirements within industry factory settings constrained by various regulations and protections. Similarly, papers X32 and X24 address the issues surrounding sensitive information and security during data collection, emphasising the need to maintain reliable and secure data sources to prevent breaches.

Paper X29 highlights that data scientists face challenges due to restricted access and varying privileges necessary to protect data privacy, significantly hindering collaboration and understanding of confidential data. Furthermore, paper X33 outlines the challenges in demonstrating AI system compliance with regulatory standards.

Industry Insight: Interview insights reveal that a significant challenge lies in complying with stringent regulations, such as General Data Protection Regulation (GDPR) while maintaining system performance. Adjustments to comply with evolving regulatory standards often lead to reduced maintainability and, in some cases, necessitate a complete platform overhaul (P14, S2). Devops Engineer P14 about system S2 elaborated on the GDPR compliance challenges: "The challenge here [GDPR rules] must all be mapped programmatically, and the rules are constantly changing."

Furthermore, the GDPR mandate to delete user data upon permission revocation introduces complexities in data management, necessitating proactive considerations in the data storage phase to streamline the deletion process (P4, S4).

C5.2 Ethics

Academic Insight: Paper X42 calls attention to the scarcity of research on how complex data cleaning methods affect AI system explainability, underlining the legal aspects. Paper X39 acknowledges the complexity of explainability as a challenge for practitioners, compounded by limited knowledge and tools.

Papers X1 and X5 touch on ethical fairness in data analysis, noting how various factors, like demographics, can influence outcomes. X28 and X9 delve into the ongoing issue of ensuring data is free from biases and has fairness in distribution. Paper X23 emphasises the importance of establishing frameworks for assessing fairness and ethics in data usage.

C5.3 Data Security

Academic Insight: Paper X1 identifies data poisoning as an emerging threat, emphasising the ease of circulating malicious data. Paper X25 highlights the need for advanced and robust data anonymisation, encryption, and access control to address the growing challenges of data privacy and security associated with the huge volumes of data and extensive collaboration practices.

Industry Insight: Participants noted the complexity of implementing robust security measures, like individual encryption keys for users, which significantly increases system complexity and management difficulty (P12, S4). The requirement for data encryption introduces additional operational overhead, posing a challenge despite its necessity for securing sensitive information (P9, S9).

P9 from S9 shared experiences with encryption: “And we had like different iterations on what type of encryption we used there [. . .] which one is currently being used that I would say can be a little bit annoying.” They further mentioned, “And of course, you also have security concerns to not have too many permissions across systems” (P9, S9), highlighting the balancing act required between adequate security measures and system usability.

C6 Data Collection

Several challenges in data collection were identified, highlighting the critical importance of data and its sources in ensuring availability, quality, consistency, and diversity. The discussion also elaborates on the impact of these challenges and the complexities involved in managing them. Additionally, it discusses the planning aspects of data collection, specifically data requirements, and discusses the associated difficulties.

C6.1 Insufficient Data

Academic Insight: Data scarcity was identified as a significant issue, with papers like X24, X1, X2, X18, X41, X43, and X39 noting the difficulty of building effective models due to the lack of sufficient, high-quality data. This scarcity often stems from the need for large datasets to yield better results in ML or deep learning models. X24 mentioned that the unavailability of data could be attributed to the diversity of heterogeneous sources, complicating the data collection process and

making it time-consuming. X1 pointed out the challenge in implementing deep learning models due to data unavailability. Concerns about data scarcity affecting ML/AI model performance were raised by X11, while X9 and X27 discussed the issue of data not adequately representing the required problem domain or system. X14 observed that reluctance to update necessary hardware or software for data collection contributes to this problem. Papers X4 and X31 highlighted issues with data that is only partially representative of the production environment or lacks comprehensive environmental information, respectively. Furthermore, papers X26 and X24 discussed the challenges posed by datasets with missing values or data that do not effectively represent the modelled process.

Industry Insight: The challenge of insufficient data in data collection and model training was highlighted across various interviews, pointing out the significant impact on machine learning and artificial intelligence models. Practitioners noted the struggle to find valuable data, often hindered by noisy data that lack true positive signals, affecting the accuracy of model predictions (P9, S9). "[...] and then I would say there's a bit of a challenge that the true positive signals that you have are not noisy, that you really make sure, OK, these are good and pure," P9 about system S9 mentioned, underlining the necessity of having clean and accurate data for model training. P7 about S7 also described the process of obtaining ground truth data as lengthy, emphasising the difficulty of starting from scratch to reach a point where predictive analytics becomes feasible: "how to get our own ground truth data so starting from having no data to actually being able to do. . . prediction is quite a long way." Also, due to unavailability or incomplete data, often functionalities and models are altered to suit the available data which is accessible(P8, S8). Product owner P4 from S4 also highlighted the dependency on data availability for success, stating, "[...] if you don't have the data, it won't help you there." It is also difficult for a particular data source to fulfil all requirements, and then other sources need to be explored and integrated (P9, S9).

The challenge of insufficient data extends to dealing with incomplete information from external providers. As data engineer P11 from S12 articulated, "There's sometimes you only get average values from API, and you know that they have the data on the way how to calculate the average because they need to calculate it in their systems, but you don't get the raw values to calculate your averages in your system when you want to have it so that that's also a problem." This specific example highlights the complexity of the issue and the challenges it poses in data collection and requirements. Similar to academics, it was found that missing data values lead to assumptions and time consumption in understanding the data. Also additionally, much time is consumed in understanding the reasons behind the missing data(P2, S1)(P1, S1)(P6, S3). Data analyst P1 about system S1 shared experience of debugging missing gaps in data, "[...] in the first few months we did not know whether this is like the clear trend that you are always missing 5 to 10% or is it like something wrong with the data and so on."

Furthermore, real-time pipelines are expected to deliver comprehensive and processed datasets. However, failures in these systems often lead to data loss, mainly when the need arises to reprocess historical data beyond the retention period of the source systems (P11, S12).

C6.2 Heterogeneous Sources

Academic Insight: Paper X8 highlighted the complexity of collecting data from varied sources, such as mobile devices and browsers. Paper X12 discussed the difficulties in integrating, merging, and accessing data from multiple sources. Paper X24 pointed out the primary challenge of harmonising data collection and discovery across diverse network systems, hindered by non-standardised vendor solutions and the complexity of sourcing and refining data.

Industry Insight: Integration of data from heterogeneous sources was cited as problematic, with the difficulty of bringing together data from distributed and diverse systems into a coherent format and ensuring all requirements are fulfilled (P9, S9)(P7, S7)(P3, S2). Product owner P3 about system S2 described, "Putting that all into one place is very hard for us because you have the systems which are just so distributed and different."

C6.3 Data Source Quality and Inconsistencies

Academic Insight: The stability of data sources was flagged as a concern by papers X24 and X26, with both mentioning the lack of reliability due to external data providers. Paper X3 shared experiences of discarding AI models because the data did not meet the required quality and quantity standards, underscoring the challenges in determining what constitutes high-quality data and understanding the specific data needs for model development. Paper X15 underscores data source issues, such as sensor problems, as contributors to poor data quality, stressing the need for timely detection and resolution.

Industry Insight: Challenges with data source quality and consistency were prevalent, with issues such as changing APIs, data unavailability, and inconsistencies causing instability and broken pipelines, as reported by (P4, S4), (P6, S3), and (P9, S9). Data infrastructure engineer P6 from S3 remarked on the frequent API changes: "A lot of the times, these APIs change or are broken, so it's been kind of challenging a lot of the time." Issues like data format changes by providers can cause substantial data loss and affect pipeline stability (P11, S12)(P12, S4). The correction of these is also challenging. The data dependency on external providers can also be challenging due to communication problems and lead to delays(P6, S3)(P12, S4). Practitioners also mentioned problems with poorly documented APIs, leading to cumbersome data extraction processes (P4, S4)(P11, S12). As data engineer P11 about S12 describes: "So basically one of the problems is that API and stuff like this are not so good documented and sometimes the process to get data out of the system is complicated." If the API lacks start process indicators and status queries, it creates uncertainty during long-running processes (P11, S12). API sources also often have some upper limits set by the data source provider, and keeping track of them is challenging, leading to errors (P13, S10)(P11, S12). "[...] API is not responsive, [and it is] not working. You're running in some API limits." P11 expressed about system S12. Collecting data from an external source becomes challenging when data providers do not provide an API endpoint (P1, S1).

C6.4 Data Requirements Planning

Academic Insight: X3 highlighted that a common issue leading to insufficient data for AI models is that organisations need more foresight and awareness regarding the data requirements needed to meet business goals using AI.

Industry Insight: Interviewees expressed difficulties defining what data is needed to meet business requirements (P1, S1)(P15, S4). P15 stressed the need for early consideration of data use cases in system S4: "And what I see kind of missing, still missing at some point is kind of like thinking at the beginning about concrete use cases or data use cases."

The initial lack of focus on specific use cases and business objectives leads to inefficiencies and unclear data collection goals. Product Owner P3 from S2 mentioned the underutilisation of collected data due to a lack of awareness of its potential applications: "So like the actual usage of data is often unknown to the users or it is often not known what can be done with this huge amount of data that they are collecting." "Also, [for future AI applications], the data should be properly processed, as the lack of data availability makes it impossible" (P4, S4).

C7 Skills

This category encompasses challenges related to the development and enhancement of skills, as well as the acquisition of talent possessing necessary competencies within the industry.

C7.1 Skill Competency

Academic Insight: X22 and X43 highlighted the industry's struggle to find skilled professionals in machine learning and data engineering, a challenge also acknowledged by X4, who pointed out organisational difficulties in sourcing appropriately skilled talent.

X6 delved into the industry perception of a deficit in data-oriented solution mindsets, attributing this to a combination of data illiteracy and inadequate documentation. These factors contributed to a broader lack of understanding regarding the potential applications of data and AI in problem-solving. X11 emphasised the role of insufficient data literacy training in the subpar skill levels of practitioners working with data for AI applications.

The issue of skill competency extended to data scientists, particularly those with backgrounds different from software development, who found it challenging to maintain their skills, as discussed by X13. Additionally, X10 identified a general lack of knowledge in the machine learning domain across various roles and profiles, which complicated collaborative efforts and limited the effective use of ML in diverse use cases.

Industry Insight: Skill competency in adopting new technologies faces hurdles related to team maturity and the availability of technical guidance. For instance, P12 about system S4 expressed, "One challenge is definitely a pretty young team. So the team still needs to learn and find their own things [. . .]," pointing out the learning curve for less experienced teams. Meanwhile, (P4, S4) noted the difficulty in onboarding different teams to new technologies: "We're more platform team and not a service team [...] it's a challenge to teach because you're still busy doing a job."

When onboarding new tools or technology, P1 and P2 in system S1 experienced challenges such as learning the best practices and trouble with debugging. Furthermore, the lack of documentation and reliance on verbal knowledge sharing makes it tough for newcomers to acquire necessary technical guidance, underscoring the importance of formal knowledge transfer mechanisms (P9, S9)(P2, S1).

The necessity of strategic training to develop proficiency is evident; however,this involves significant organisational time and resource investment (P12,S4).

MLOps tools and topics like versioning are generally introduced at later stages of a machine learning system's maturity due to their complexity and novelty (P9, S9)(P7, S7). The challenge also encompasses the substantial time commitment to identify effective utilisation and process establishment (P10, S11). Advanced technology management, such as managing Kubernetes, requires significant expertise. P14 about system S2 states, "Kubernetes is quite complex, and that's why we have made all our certifications," highlighting the need for advanced skills and certifications.

C8 Costs

This challenge category is predominantly highlighted in industry interviews rather than academic discussions, and thus, the insights are presented solely from an industry perspective. It focuses on activities contributing to escalating costs and their need for mitigation. Additionally, it examines how cost considerations serve as a limiting factor in industry decision-making processes.

C8.1 High Costs

Industry Insight: In interviews, the challenge of resource optimisation was underscored by practitioners pointing out the significant cost increases from scaling data models and executing resource-heavy queries (P5, S4)(P1, S1)(P2, S1)(P4, S4)(P12, S4)(P13, S4). P4 about system S4 highlighted the cost implications of querying large datasets: "[...] must not be used when you query a big table because it will explode the cost and take a long time."

P5 about system S4 discussed the inefficiencies leading to unnecessary costs: "[...] data democracy and everything it's really hard, people who hardly understand what they're doing are creating costs that are not necessary [. . .]" P6 about S3 identified the optimisation challenges with integrating multiple behaviour sources, while P9 about S9 discussed the high billing costs associated with large volumes of data and P8, P10 about S8 and S10 respectively emphasised the necessity of making trade-offs between model performance and computational expenses.

P8 about system S8 expressed the prohibitive cost of GPU for testing: "The problem is you need to have a GPU to test it, and then it would just cost money to explore it. I did not test it because I would need to have, [for example], a GPU instance all the time." P6 about system S4 suggested that there is room for more cost savings: "[...] we still have way too much compute running compared to what we need at any given time."

P7 about system S6 stressed the need for careful management to prevent high costs: "[...] how fresh the data is and how often we actually regenerate and what we need to regenerate without exploding the credit card," indicating the financial balancing act required in data handling.

5.4 Solution Design

5.4.1 Tools Evaluation

In our search for tools that align with open source and cost-effectiveness criteria, we evaluated several options to manage data versioning and facilitate experiment management during initial model development. Data version control (DVC) emerged as particularly suited to our needs due to its simplicity in setup and comprehensive feature set.

DVC

DVC is often referred to as "Git for data." It extends the functionalities of Git to data files, enabling version control of large datasets alongside code. Data Version Control (DVC) stores metadata and file changes in a Git repository while storing the data in separate remote storage. This setup helps track data modifications through reference pointers (hashes) stored in the Git repo, linking data versions directly to Git commits for precise version tracking and restoration, analogous to how code is managed. DVC's workflow is similar to Git's, which helps ease its adoption among users familiar with Git. For instance, `dvc add` starts tracking data files by creating a pointer file for remote storage, which is managed with `dvc push`. However, updates to already tracked files require `dvc commit` to register the changes, a process slightly different from Git's use of commit as well as handling of modified files.

Pipeline and Experiment Management with DVC: DVC's pipeline functionality is particularly beneficial. It enables the modularisation of the data processing workflow into stages, each defined with specific input and output dependencies. This structure not only ensures the reproducibility of results but also automates and streamlines workflows, reducing the potential for manual errors. Executing a DVC pipeline with `dvc repro` automates and tracks changes efficiently. Parameterisation within pipelines, useful in machine learning for running workflows with varied settings, enhances the reusability of these pipelines across projects. Additionally, this approach can promote the reusability of pipelines across multiple projects. As DVC also support experiment tracking, we will further take advantage of the DVC pipeline and use it to run the experiments(`dvc exp run`).

Comparing DVC and MLflow: While MLflow excels in experiment tracking and offers a flexible platform for managing the ML lifecycle, it requires a dedicated server for effective collaboration and experiment sharing, which can complicate initial setup and maintenance, especially for small teams. In contrast, DVC operates without dedicated server infrastructure, utilises existing storage solutions, minimises overhead, and fits well within budget constraints. This setup aligns closely with our criteria for low setup costs and user-friendly operation. Additionally, unlike DVC, MLflow does not automatically reproduce data versions when experiments must be rerun, requiring manual intervention to ensure data consistency across experiment reproductions. From a cost perspective, DVC is advantageous as it leverages existing storage solutions (like cloud buckets or networked file systems) and minimises the need for specialised servers. This aspect is particularly appealing in exploration settings where budget constraints are a significant consideration. This also further satisfies acceptance criteria for ease of use.

Conclusion

Given its seamless integration with Git, ease of setup, and efficient handling of large data and model versioning, DVC was chosen for its strategic advantages in data versioning, experiment tracking, and automation. While MLflow offers robust tracking capabilities, its setup complexities did not meet our requirements for a straightforward, cost-effective solution. To use DVC effectively, the solutions will use parameterised DVC pipeline shown as example in Figure X and integration of Git hooks to automate DVC actions which reduces overhead.

5.4.2 User Scenarios and Workflow Design

In this section, we will illustrate how the chosen tool addresses the defined problems and meets our objectives. Using user scenarios, we will clearly show the challenges and their solutions, ensuring the tool meets the established requirements. This approach includes describing typical user scenarios and workflows with insights and recommendations.

Workflow Setup

For all the workflows, we need a setup that includes a Git repository and DVC, enhanced with the integration of Git hooks using the DVC install command that automates specific DVC actions with Git commands, and configured remote storage for data and model versioning. The remote storage, essential for DVC, stores data files and model artefacts outside the Git repository, allowing for efficient data management and retrieval without burdening the Git system, ensuring seamless integration and scalability. If the DVC remote is configured to use a cloud storage bucket, additional setup for authentication and access management is required.

User Scenario 1: Data Versioning and Rollback

In data-driven projects, data often evolves over time, necessitating the creation of multiple data versions. It is crucial for data scientists to be able to track different data versions and revert to previous data versions for error correction or comparison purposes. This scenario addresses the workflow required to manage evolving datasets efficiently in a collaborative environment.

The structured approach to data versioning is presented in the workflow in Figure 5.2, using DVC and Git. This approach not only tracks changes but also links data states to specific outcomes or stages of the project. This workflow is fundamental for maintaining a record of changes and enabling the mechanism to revert to previous states, which is essential for reproducibility and data integrity in data science and machine learning. This approach also facilitates seamless team collaboration, allowing members to easily access the latest data version and synchronise with the necessary version, thereby preventing any confusion as the data evolves. The workflow requires the initial setup as detailed in the Workflow setup.

Maintaining a History of Changes: To maintain the versioning of the data, we need to start tracking it. This process can be managed with DVC and Git, where DVC tracks the data file and generates a `.dvc` file that Git then tracks. This way, the data files can be versioned simultaneously

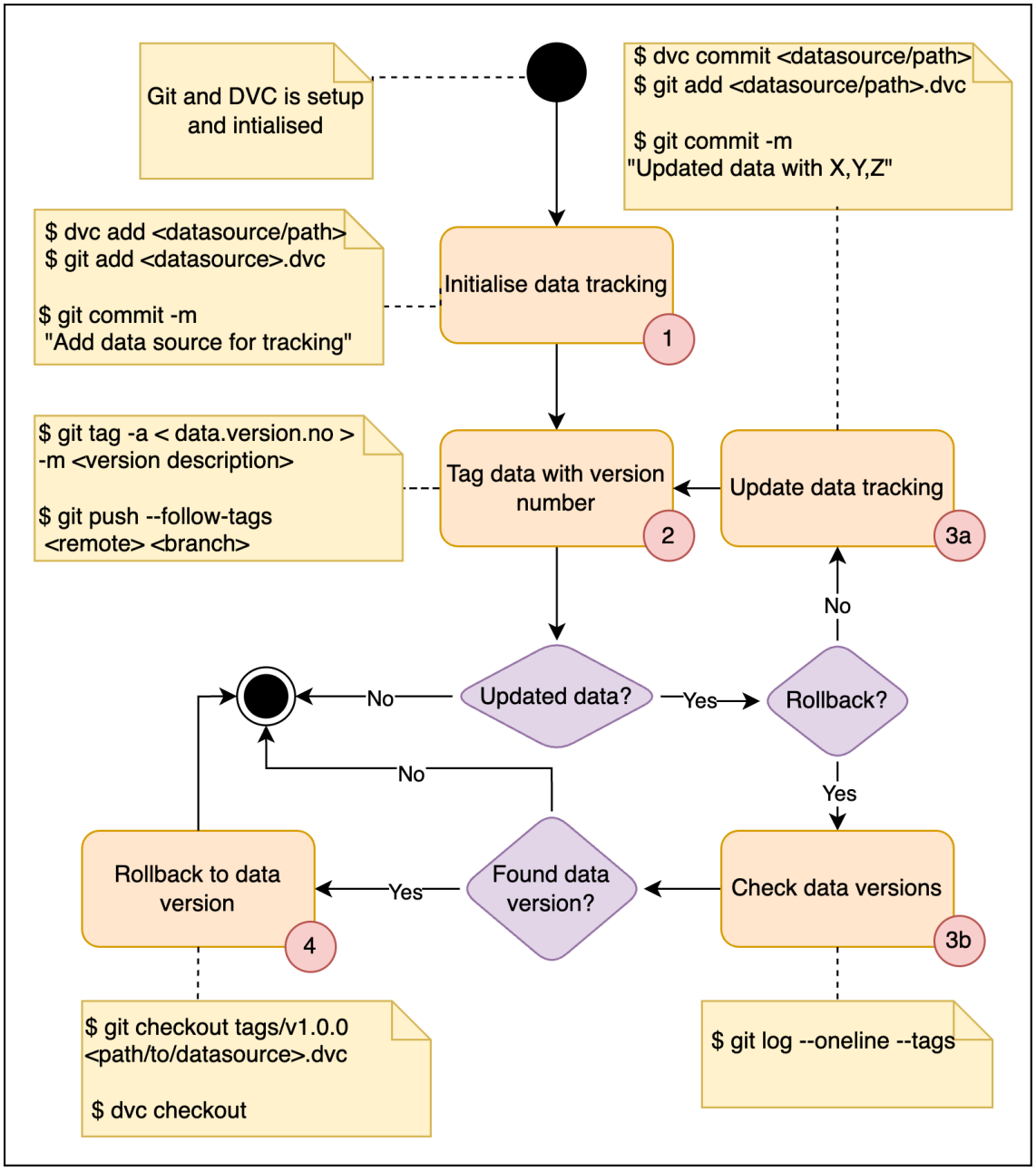

Figure 5.2: Solution Workflow for Data Versioning and Rollback

with the code, streamlining project management. Data should then be tagged with a version number, as shown in Step 2 of the workflow, which creates a clear reference point for the data's state at that moment. When data is updated, the tracking information must also be updated, and a new version tag should be assigned as outlined in Step 3a of the workflow. These version tags can be linked to specific iterations of machine learning models, enabling practitioners to correlate model performance with corresponding data versions. This systematic versioning is essential for clear team comprehension and maintaining good documentation. However, the standardisation of versioning semantics requires further exploration and varies based on factors like the data type, project nature, and the versioning practices adopted by the team.

Reverting to a Previous State: Data corruption and introducing errors during updates are not uncommon in data-intensive projects. When such issues arise, the workflow provides a rollback mechanism in Step 3b, Step 4, restoring the data to a known good state, verified and error-free. This rollback capability supports robust data management by minimising the risk associated with data updates. When new data potentially corrupts the dataset, the workflow facilitates a quick restoration without manual data cleansing or correction, saving time and reducing the likelihood of human error.

Key Takeaways

- **Efficiency with Large Data Sets:** If you're dealing with very large data files and want to avoid duplicating storage in the local DVC cache while still tracking the data with DVC, you can use the `--to-remote` option with the `dvc add` command. This option allows you to bypass the local cache, storing the data directly in remote storage.
- **Remote Storage Compatibility:** If data needs to reside in remote storage (such as HTTP, S3, SSH, etc.), you can track it using DVC by maintaining only the DVC tracking file in the repository. This is achieved using the `dvc import-url --to-remote` command. To ensure that the tracking file remains current, you may need to execute the `dvc update --to-remote` command.
- **Branch-specific Data Handling:** When switching between branches, DVC updates the workspace data to match the versions specified in the `.dvc` files committed to the target branch. If the required data versions aren't locally available, DVC pulls them from remote storage. However, updates are not automatic for data files stored in external storage. Manual intervention is necessary, as DVC does not manage external storage directly to prevent data loss.
- **Granular File Tracking:** DVC tracks changes at the file level, meaning that it saves a new file copy to storage whenever you change it. If you modify just one line in a large file, DVC stores an additional full copy of that file. However, when tracking a directory with many files, DVC only stores a new copy of the files that have been changed, not the entire directory.
- **Directory Tracking:** In case of tracking multiple data files in the data source, track the directory(s) instead of each file. This will keep the number of `.dvc` files low. For instance, it's more practical to manage datasets with a single `dvc import {url} data_dir` for the entire directory rather than multiple imports for each file, simplifying reuse and reducing complexity.

User Scenario 2: Experiment Tracking and Collaboration

Data scientists aim to develop and tune models through experimentation. This requires running multiple experiments, adjusting parameters, and collaborating with peers. The challenge lies in managing these experiments efficiently and ensuring seamless collaboration among team members.

Figure 5.3 illustrates the workflow to facilitate systematic experimentation, supporting the tracking, evaluation, and enhancement of experiments within a collaborative setting. This allows experiments to be shared, reviewed, and iteratively improved upon. It introduces the potential for integrating continuous development and integration processes, which enhance the project's reliability and efficiency. The workflow requires initial setup as detailed in Workflow Setup.

DVC Pipeline Setup: As illustrated in Step 1 of the workflow, the DVC pipeline with parameterisation is set up to automate the workflow and track the output files from various processing and model development scripts. Parameterisation in the DVC pipeline's YAML file enables flexibility and simplifies adjustments by allowing users to easily configure and reuse settings across various scenarios and projects, streamlining development and testing processes. Furthermore, the DVC pipeline's systematic approach ensures that only the stages with changed dependencies are executed. This minimises resource wastage and optimises computing power and storage during model testing and development.

Experiment Tracking and Evaluation: Before initiating experiments, it's crucial to track all relevant files as outlined in Step 2 of the workflow. Then, use the DVC pipeline to run and monitor experiments as shown in Step 3. This approach allows for executing multiple scripts through a single command, streamlining the management and automation of complex workflows and facilitating minimal setup for initiating and running experiments with little disruption to the existing codebase. This setup supports experiment reproducibility and transparency. To review the results, utilise the DVC command line as per Step 4 or the VS Code plugin. This review is an essential checkpoint to evaluate the outcomes and decide on their validity, guiding decisions on further development or sharing of the experiments.

Tune Experiment Parameters: After evaluating the experiments, if the results suggest adjustments, proceed with refining or iterating the experiment through Steps 5a or 5b. The choice between these steps depends on whether to run multiple experiments with varied parameter values using the queuing method in Step 5b or proceed with a single experiment in Step 5a. Opting for queuing is advantageous when handling a large volume of experiments to manage computational resources and time efficiently. This process of tuning parameters may require the iterative testing and review of various hypotheses.

Collaborative Environment: When experiments yield substantial results, they should be preserved to facilitate reproduction across different environments. Substantial experiment results encompass successful experiments, which directly contribute to advancing the project, and the insights gained from failures, which can be invaluable for refining future research efforts. This practice of persisting experiments is crucial for conserving resources and safeguarding knowledge. To persist experiments, they should be pushed to the Git repository, as indicated in Step 6 of the workflow.

Moreover, experiments that are successful or require further development or collaborative efforts should be isolated in their own branch. This can be easily achieved by DVC, which takes all the files related to the experiment and creates a git branch of the experiment as recommended in Step 7. This

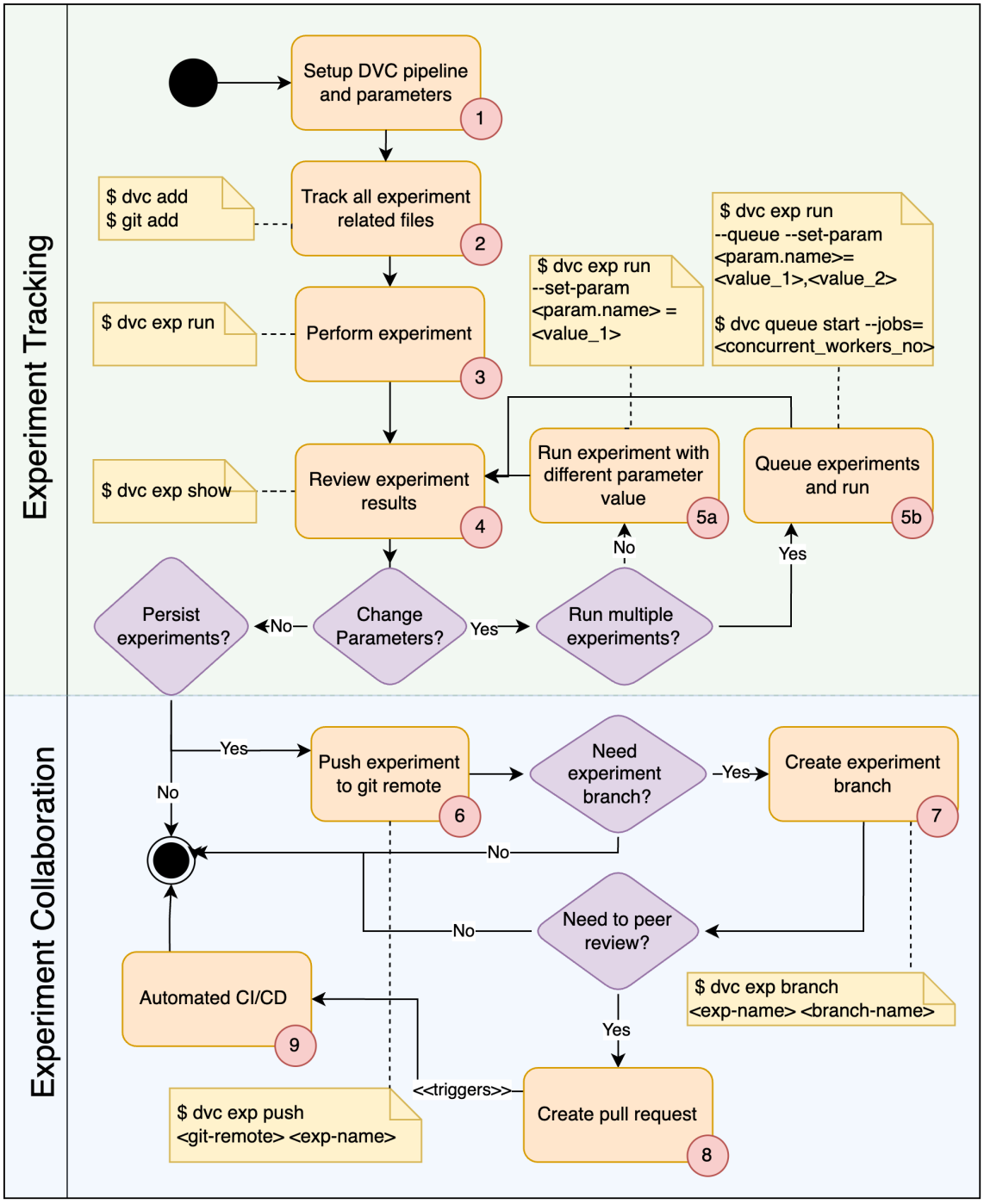

Figure 5.3: Solution Workflow for Experiment Tracking and Collaboration

isolation in a dedicated branch simplifies collaboration by enabling team members to easily access, review, and extend each other's work. This approach not only enhances collaborative efficiency but also supports the detailed tracking of changes and ideas.

When an experiment proves successful and requires peer review, as detailed in Step 8, a pull request from that branch should be created. This peer review process is essential, particularly when the experiments have significant implications for the main project or when peer validation can improve the research's quality and reliability. It also serves as a quality control step, ensuring that the experimental modifications align with the project's objectives and quality standards.

In addition, as described in Step 9, the pull request can trigger CI/CD automation processes, such as testing the experiments and appending comments to the pull request with performance metrics comparisons. After the pull request is approved and merged, the model can be added to the registry, finalising the integration of successful experimental outcomes into the broader project framework. This streamlined integration aids in maintaining a continuous cycle of development, testing, and enhancement, facilitating ongoing improvement and innovation in the project.

Key Takeaways

- **Track files before running Experiments:** Ensure that all code and data files are tracked before running experiments. This practice is needed since experiments will only save the changes in the files being tracked by DVC or Git.
- **Continuous Data update:** Set flag 'persist: true' in DVC pipelines for models and datasets that require frequent updates. This setting prevents the DVC pipeline from deleting and rebuilding outputs from scratch in each run. This enhances efficiency by maintaining the state of critical data and model outputs across runs, allowing for continuous data integration without the overhead of reconstructive processing. However, this configuration is not ideal if reproducibility from scratch is required.
- **Visual Experiment Management:** Utilise the DVC Visual Studio Code plugin or DVC Studio (a web-based application) for a visual interactive interface, enhancing the visibility and manageability of experiments across the team.
- **Integration with Advanced Tools:** DVC supports hyperparameter tuning tools such as Hydra for advanced parameter management, enhancing the flexibility and scalability of the experimentation framework.
- **Advanced Monitoring metrics:** Integrate DVC Live python library to automate logging system and performance metrics and support real-time monitoring. This tool also aids in visualising results, making it easier to analyse and interpret experimental outcomes.

User Scenario 3: Experiment Reproducibility

Given the complexity of conducting numerous experiments involving varied codebases, parameters, and data versions throughout a project, what workflow should be followed to enable team members to efficiently revisit and reproduce these experiments, ensuring that results can be consistently verified and development efforts can be seamlessly continued?

The workflow depicted in Figure 5.4 offers a structured and reliable method for replicating experiments. It confirms that experiments can be consistently reproduced across various configurations—including data, code, and parameters—thereby affirming the stability and reliability of the model development process. This workflow supports experiments that can be reviewed both locally and remotely. It allows data scientists to either confirm previous results or set the foundation for further exploration. The workflow emphasises the importance of a shared experimental setup, where findings can be reproduced, validated, and extended by team members. The workflow underscores the value of a shared experimental setup, where findings can be reproduced, validated, and extended by team members. This workflow assumes that the experiments have been conducted using the solution described in Scenario 2. To start this workflow, the repository should be cloned into the user's workspace.

Experiment Review and Retrieval: With DVC and Git, experiment tracking ensures that experiments can be displayed and reproduced if present in the local workspace. By default, experiments are saved locally, but for broader reproducibility, they must be pushed to the Git repository. If experiments are unavailable locally but needed for comparison or reproduction, proceed with Step 1b, which involves checking the remote repository and pulling experiments into the local workspace.

When the experiment is available in the local workspace, proceed with step 1a to verify its presence. If the experiment is not found locally, then it can be searched for in the remote repository. This process may involve an iterative cycle of searching until the experiment is located or the search is concluded if no suitable experiment exists.

Experiment Reproduction: Once the desired experiment is identified, the process moves to Step 2 for experiment reproducibility. Here, the original experiment's exact parameters and conditions are reapplied, and the experiment is rerun. This step is crucial as it confirms the experiment's reproducibility, ensuring that results are reliable and verifiable.

Key Takeaways

- **Managing External Data:** For experiments that use data from external storage, it is required to synchronise these files manually. So after the experiment environment is applied, only the `.dvc` file will update, and data needs to be pulled from the remote storage and uploaded to the external location. This is similar to the rollbacks of the external storage data.
- **Optimise Resource Usage for Efficiency:** Reproducing experiments, especially with large datasets, can consume substantial computational resources and time. To address this in scenarios where reduced time and computational overhead are crucial—such as during local development or resource-limited conditions—consider employing a dedicated branch that contains a smaller subset of data. By pulling this data and combining it with the configuration of other experiments, you can significantly lower resource usage and expedite the experimentation process, making it more manageable and efficient.

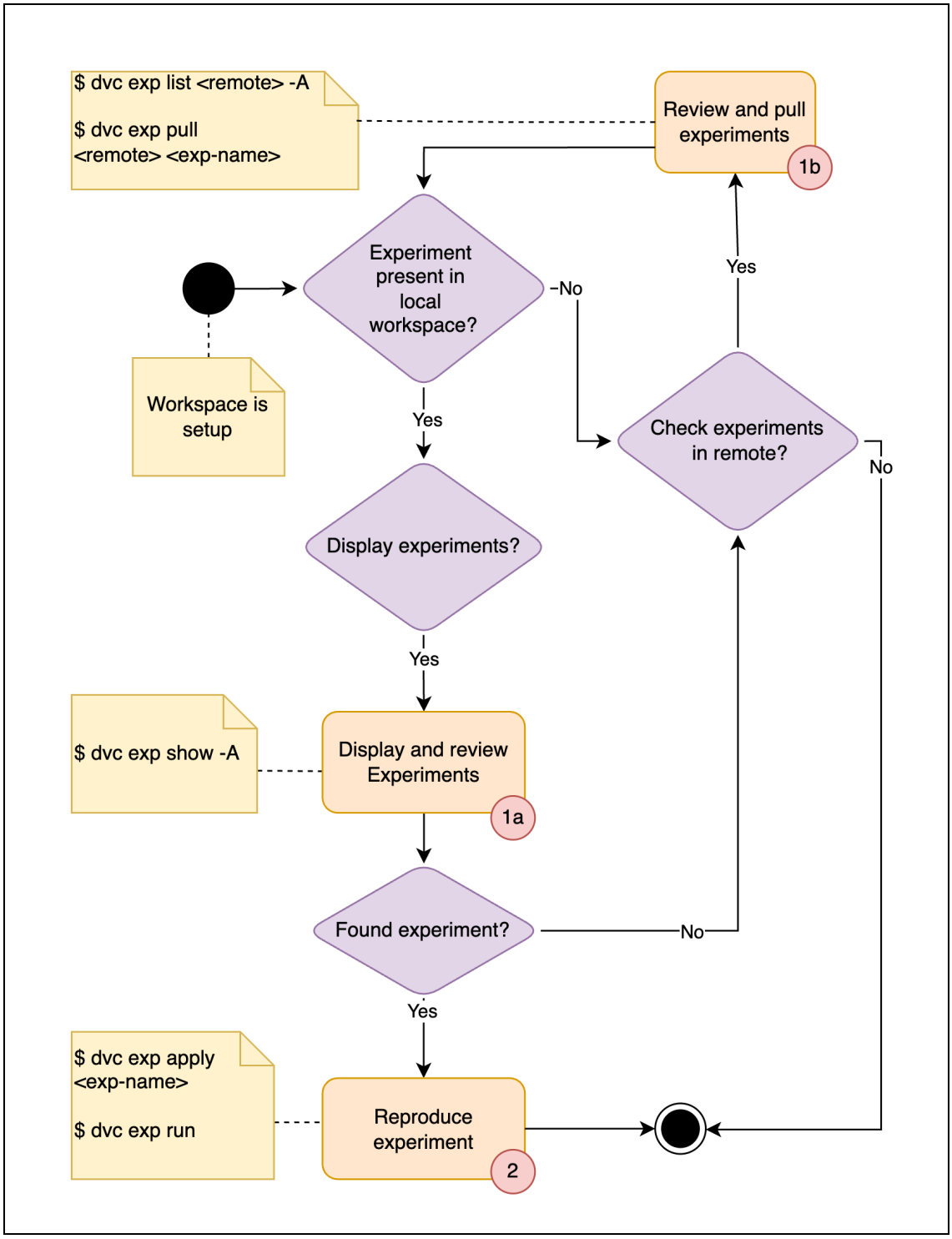

Figure 5.4: Solution Workflow for Experiment Reproducibility

6 Discussion

In this chapter, we will discuss our findings, focusing on a comparative analysis of challenges encountered in academic versus industry settings and exploring the underlying reasons for these differences. To facilitate this comparison, we have Figure 6.1 depicting data pipeline and operation challenges,Figure 6.2 illustrating data understanding and collaboration issues, Figure 6.3 showing data collection difficulties, Figure 6.4 focusing on challenges in data preparation, Figure 6.5 representing responsible data management challenges and Figure 6.6 presenting challenges in data monitoring and testing. These figures contain bar graphs for each challenge label within the category, compared between academic and industry sectors, with percentages indicating the frequency of occurrence of each challenge label within that category. This visual representation highlights which challenges are more or less prevalent in each sector and will support our discussion. Additionally, we will evaluate the outcomes of our proposed solutions, discussing their limitations and suggesting directions for future research. Finally, we will address the potential threats to the validity of our study, ensuring a comprehensive reflection on all aspects of our research findings.

6.1 Challenges in Academia vs. Industry

The design and integration of pipelines emerged as a significant challenge in the industry, attracting more attention from academia. It primarily focused on selecting appropriate tools, their integration, and the standardisation of design practices. Concerns about vendor lock-in, which creates friction when switching tools and hinders communication, especially in multi-cloud or hybrid setups, were emphasised in industry contexts. These concerns were less prominent in academic discussions, potentially because academic environments may not face the same practical constraints and commercial pressures as industry. In industry, project lifecycles tend to be longer, with a need to integrate multiple systems and ensure their integration possibilities for future projects. The need for commercial use, enterprise licences, robust support, and system durability significantly influence tool choices, distinguishing industry preferences from those in academia.

Versioning posed another major challenge in the industry, similar to results from academic studies, particularly relevant to almost all data science case studies. Since versioning data is not part of traditional software engineering, the lack of standardised practices and the time required to understand the tools and make informed decisions further complicate this issue.

However, pipeline design has not been as pressing a concern in the industry as it has been in academia. This discrepancy is partly because the issue has largely been resolved in practice, as noted by practitioner P11: “At the moment, not because we have a known working deck that is working OK, but it could be that depending on the tool decisions you make, it [could become] a huge problem.”

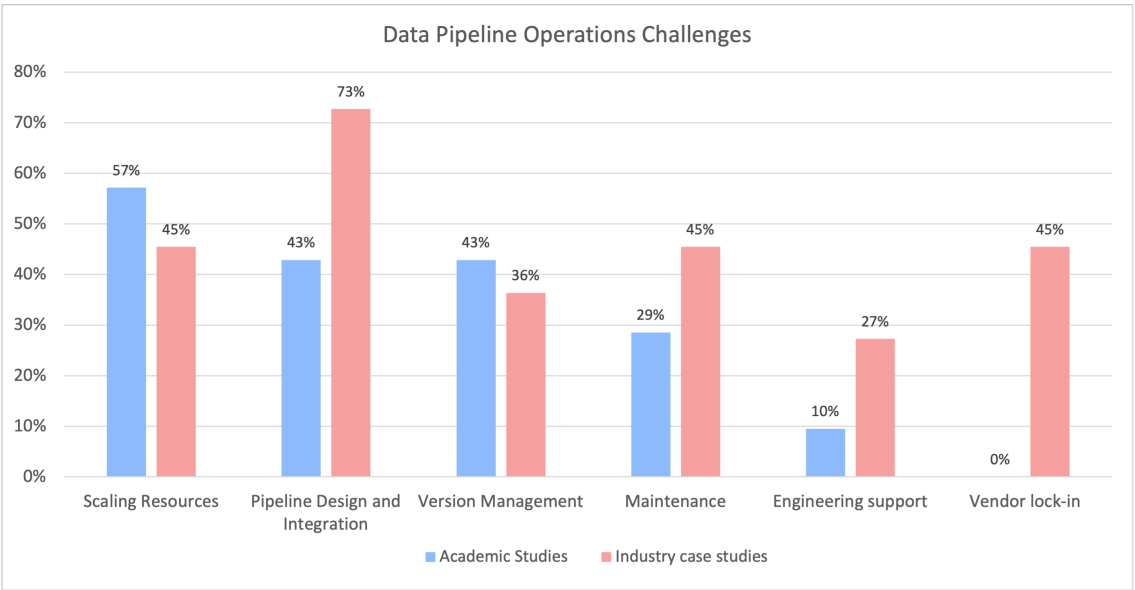

Figure 6.1: Comparative Analysis of Data Pipeline Operations Challenges

Interestingly, data scientists in case studies have emphasised the need for better engineering support for data pipeline management, a topic that has not been prominently featured in academic research findings.

Data understanding and collaboration are ranked as the second highest concerns in both academia and industry, highlighting the need for the development of practices and tools that facilitate easy access, collaboration, and knowledge sharing. However, the challenge of data understanding is less pronounced in industry than in academia. One reason for this, as explained by data scientist P10, is that data understanding is manageable even with limited data documentation if there is a contact point available to clarify the data. P10 noted, "So if the people want to do this, then they are very helpful and happy to explain [...]." Practitioner P7 mentioned that communication challenges primarily occurred with external partners, not within the company itself, stating, "[...] the communication difficulties came between us and the the partner. But inside the company? Not really." On the other hand, domain expertise is crucial in understanding data-related challenges, especially in addressing niche use cases unfamiliar to those without relevant background knowledge. Product owner P13 emphasised the need for subject matter experts, "We need experts to evaluate situations beyond my limited knowledge. This also underscores our desire to collect chat histories and process them in a way that's more digestible for human moderators or subject matter experts." Additionally, the challenge of data understanding might be influenced by expectations, as data scientist P8 remarked, "It was expected and not that challenging." Data access management remains an issue, although efforts are being made to systematise and ensure traceability of access, particularly to platforms like Snowflake via access management systems. According to product owner P5, "We have built for this now where we define the accesses, this is mainly to the snowflake platform via an access management systems where we put everything that's needed within code within YAML files, [allowing us to provide access] pretty fast and in a documented instructed way."

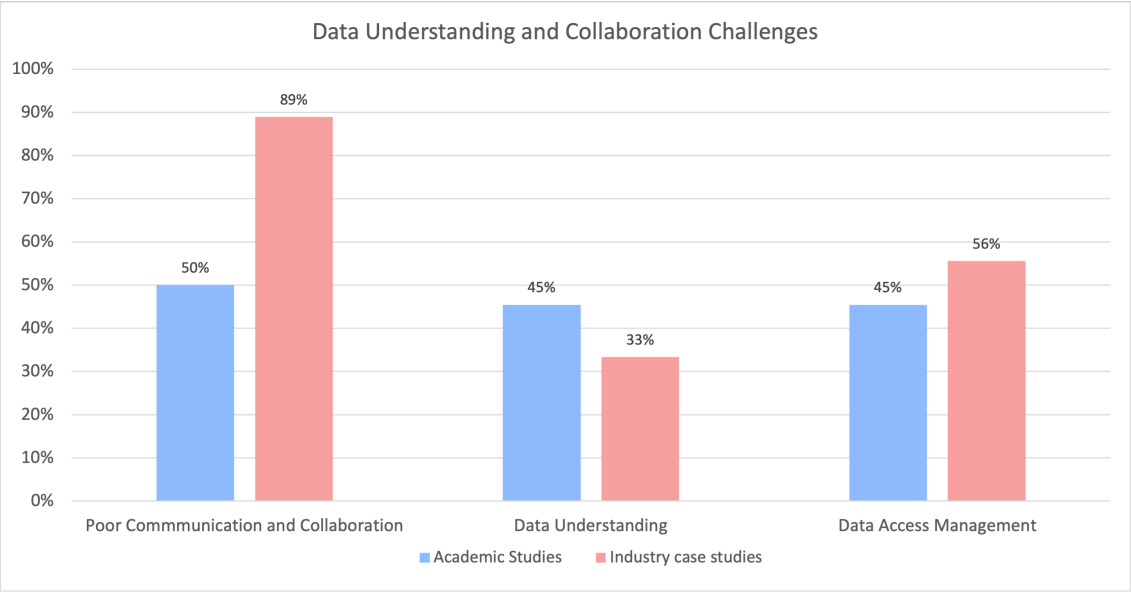

Figure 6.2: Comparative Analysis of Data Understanding and Collaboration Challenges

Contrary to academic observations, data collection was ranked second due to the poor quality of data from sources and inconsistencies within these sources, posing significant challenges for practitioners. Academics highlighted issues with insufficient or missing data relevant to business use cases, which also resonate in the industry. However, the industry's perspective is further complicated by the integration of multiple, including external, data sources and the constant evolution of these system components. Challenges such as dealing with unreliable external data sources are exacerbated when it is impossible to address issues at the data source directly or when communication about changes from the data source end is ineffective. Additionally, quality assurance with external sources cannot always be controlled, and ensuring synchronisation between organisations remains a significant challenge. These dynamics often reflect real-world scenarios that academic studies may overlook. Furthermore, the industry faces challenges in anticipating data requirements, which impacts future planning and decision-making related to data that could support future use cases. These aspects have not been as thoroughly considered in academic research.

Data preparation is ranked third in industry case studies compared to its top ranking in academia. Interestingly, concerns about data processing or management were raised by only one practitioner despite being a primary concern in academic discussions. Data preparation and related transformation challenges are discussed more extensively in academia and less so in industry interviews, as indicated by practitioners, because these challenges are typically anticipated and often already resolved, or there is a clear understanding of how to address them. Existing pipelines are functional and provide solutions, but how this prepared data can be used to achieve business goals is currently shown as a bigger concern in interviews. As practitioner P4 mentioned, "The pipeline and the infrastructure, like everything which ingests the data and transforms it, that's not a big deal; it's really working stable now more or less, but the actual consumption of the data is causing the biggest like thinking and discussions currently". Additionally, the lack of emphasis on real-time data preparation can be attributed to its lesser necessity in projects, as highlighted by P11, who stated, "Right now, you're not dealing with real-time data." Similarly, data labelling challenges were not prominently

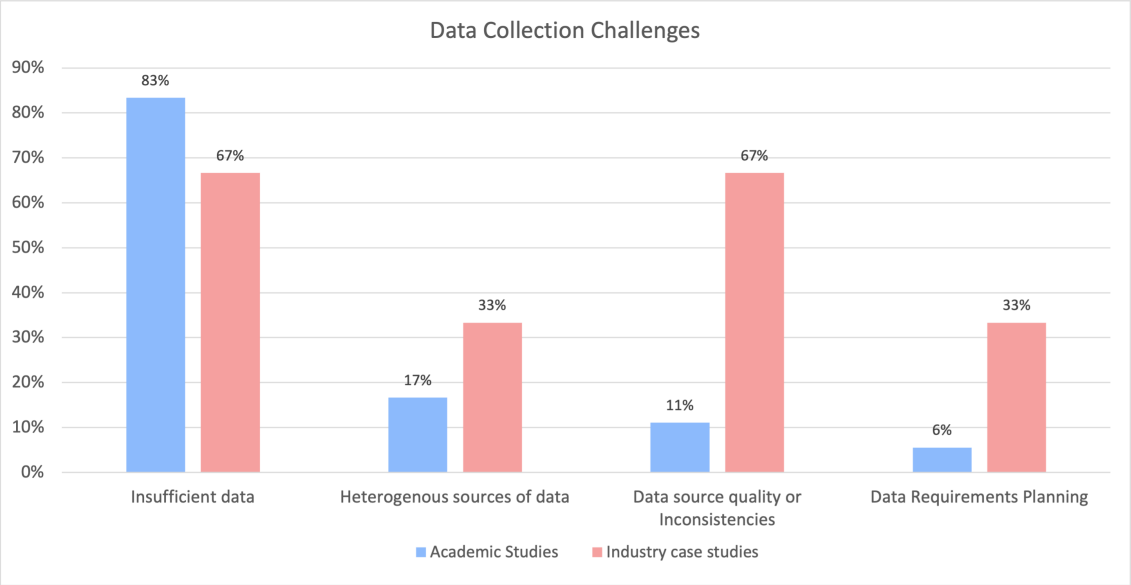

Figure 6.3: Comparative Analysis of Data Collection Challenges

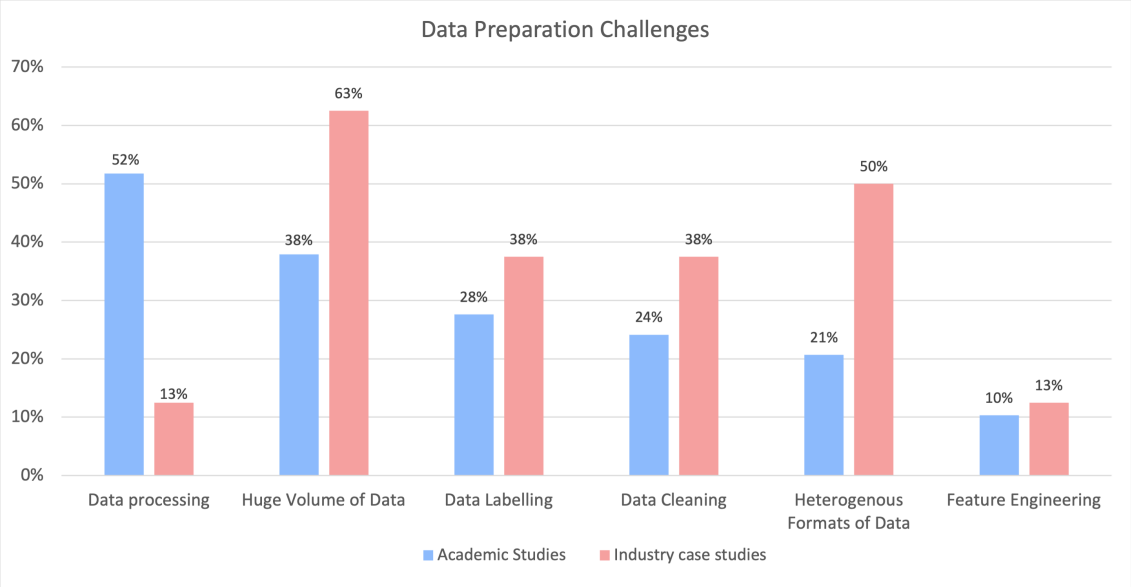

Figure 6.4: Comparative Analysis of Data Preparation Challenges

featured in interviews with practitioners, mainly due to the nature of the projects discussed, which did not require data labelling. However, data scientists with experience in data labelling expressed concerns about it and noted that projects were designed to avoid manual labelling or to make processing possible without labels. P8 described this approach: "[...] the idea was like to have more self-supervised training."

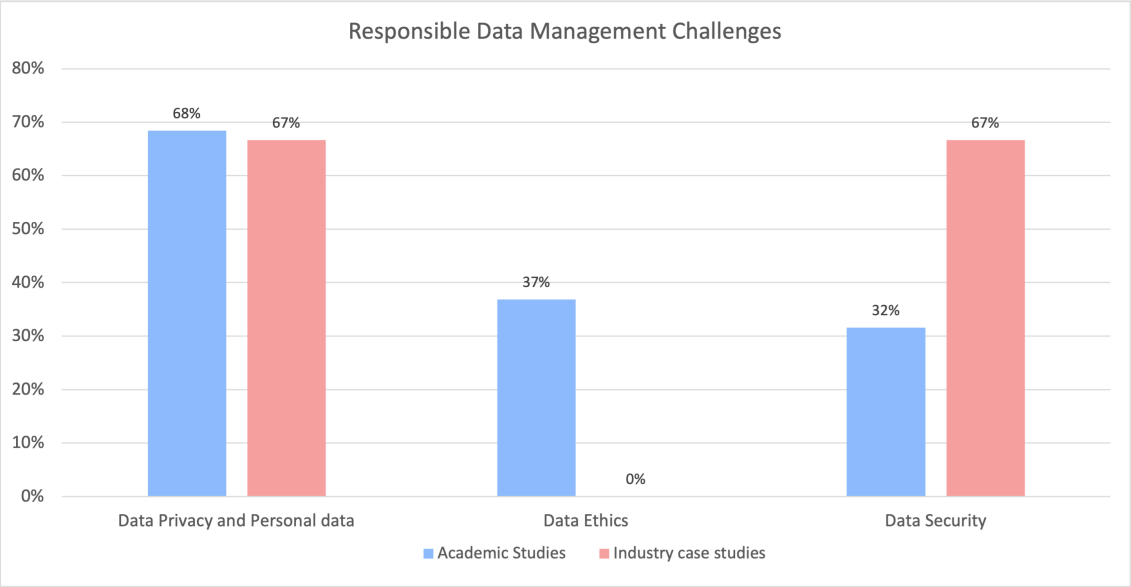

Figure 6.5: Comparative Analysis of Responsible Data Management Challenges

Challenges related to skills competency and expertise are more prominently highlighted in the industry than in academia. In academia, the focus is often on talent scarcity or the difficulty in finding adequately skilled personnel. In contrast, the industry emphasises the need to provide technical guidance to help professionals acquire or update their skills.

Cost is another significant factor discussed more in industry interviews than in academic settings. In industry, high costs are viewed as a direct challenge that influences decision-making in the design and development of solutions. In academia, however, costs are frequently mentioned as a secondary concern, merely a side effect of other challenges. This difference is understandable from a commercial standpoint and offers valuable insights for the further development of solutions.

Responsible data management, focusing on data security, privacy, and adherence to ethical and legal guidelines, was not as prominent a concern in industry case studies as it was in academia. Several reasons may have contributed to this. Firstly, robust processes are already in place within the organisations featured in the case studies, leading to fewer issues with personal identification data due to a well-established guideline system that, although time-consuming, sets clear expectations. P15 explained, “There’s a process you must go through. Yeah. And that’s for the security team, and they have to give their check. Then there’s the data privacy team, and they have to give their check. [...] It’s really highly regulated. It has some trade-offs, but in the end, when you go through the process, you can be pretty sure that you are aligned with the laws.”

Secondly, personal identification (PI) data is deliberately avoided in projects as much as possible to reduce complexity and risk. P13 noted, “We don’t want to have [personal identification] data in there. So that’s something we need to avoid.”

Thirdly, the nature of some of the case study systems does not require dealing with privacy-sensitive data. The data being used typically lacks any personal identification information. P10 stated, “So no [personal identification data], at the moment; for example, the data we are using, and this is all the data which is also online.”

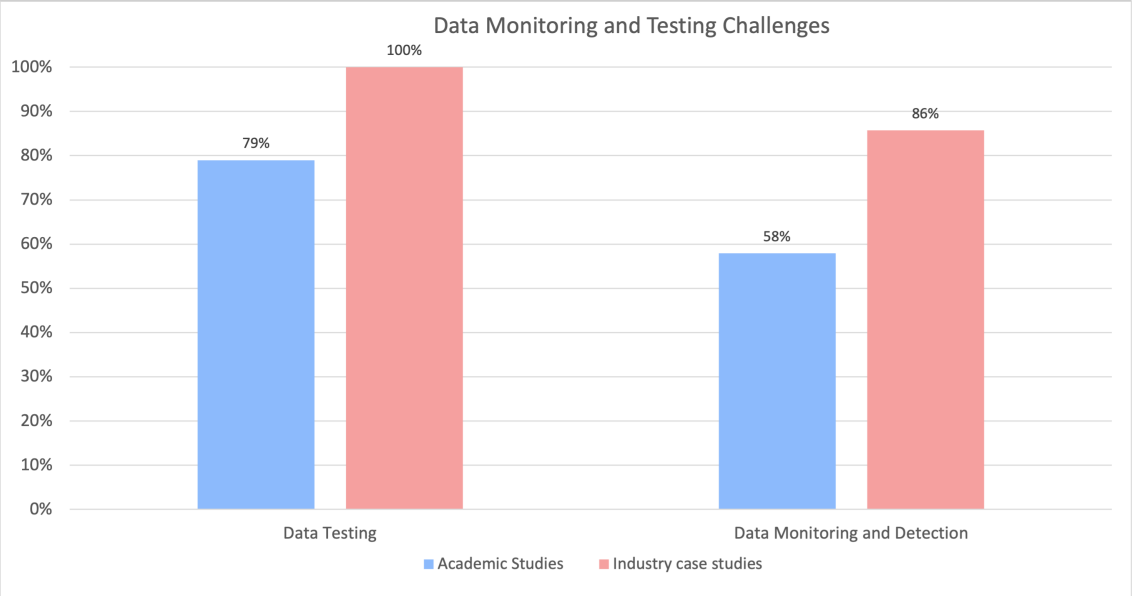

Figure 6.6: Comparative Analysis of Data Monitoring and Testing Challenges

Data monitoring and testing have garnered similar levels of concern in both academia and industry. Some noteworthy measures and trends have been highlighted to address data quality assurance and enhance system robustness in the industry. First, given the impracticality of writing tests to cover all possible scenarios, tools for data observability, such as Monte Carlo, along with AI technologies for early error detection, are being explored in various projects. Another significant development is the shift of responsibility for data preparation and quality assurance towards the data consumer. P5 explained, "We simply say the data provided to us, we can transform it if necessary in a certain way and then provide it to you. If there are any data quality issues, you need to solve them yourself." This shift highlights both the pros and cons and represents an interesting finding that could help evolve solutions.

6.2 Assessment of Solution

The workflow scenarios validated the proposed solution of utilising DVC integrated with Git effectively addresses key challenges identified from industry case studies, such as version management, reproducibility, and collaboration, while also enabling automation in the defined case study scenarios of Model development. This DVC tool successfully meets a broad range of acceptance criteria, specifically 1.1-1.9 for data versioning, collaboration, and rollback functionalities, as well as 2.1-2.12 for experiment tracking, reproducibility, and further collaboration aspects. Notably, the overlap between acceptance criteria 1.4-1.9 and 2.7-2.12 illustrates that DVC effectively serves multiple functions, simplifying toolset requirements and emphasising its broad applicability, thus demonstrating its extensive range of capabilities. However, it is important to note that these acceptance criteria (1.4-1.9) are derived from the practical needs expressed by practitioners during interviews, addressing potential challenges such as integration, engineering support, cost, vendor lock-in, open-source benefits, and robust support and documentation of the tool. This suggests

the relevance of these criteria in selecting tools aligned with industry requirements to ensure the solution's applicability and potential to address real-world challenges effectively in various scenarios.

6.2.1 Outcomes and Limitations

DVC is straightforward to set up, does not require a dedicated server, and is open-source, requiring no licensing fees, making it cost-effective. It leverages existing infrastructure and optimises computational efficiency by avoiding unnecessary re-executions and utilising cached results. DVC's integration with Git supports seamless documentation of data and experiments alongside code changes, offering a comprehensive view of project evolution. This integration makes DVC particularly user-friendly for those already familiar with Git. Although there is a learning curve associated with DVC, the benefits of enhanced team collaboration and version control justify the initial investment in learning, as highlighted by practitioner feedback. The automation from DVC's pipeline and integration of DVC commands via git hooks minimise the tool's overhead by streamlining the workflow with fewer commands, addressing issues related to multiple commands and the potential for errors, as highlighted by practitioners. Additionally, the systematic approach to running experiments using the DVC pipeline ensures reproducibility and operational efficiency by only executing what is necessary, thus reducing the risk of resource wastage, which practitioners have found beneficial. The solution supports experiment versioning, logging, reproducibility, and team collaboration by tracking all data, code, and artefacts, creating an efficient experimental setup. This setup does not require a server, further enhancing its cost-effectiveness. If a visually interactive interface is required, DVC Studio can be set up, requiring some effort. Still, alternatives like using the command line or the DVC Visual Studio Code (VS Code) extension are available to utilise most of the features. This flexibility is particularly valued at the project's onset, reflecting the need for engineering support highlighted by data scientists who prefer to focus on their experiments and models rather than on operational challenges.

The integration of DVC with Git simplifies moving between different data versions. It involves updating the `.dvc` file tracked by Git, followed by commands like `dvc checkout` or `dvc pull`, streamlining the version management process. This process is straightforward when data is located in the DVC repository but can be challenging for data stored in external storage. Even though DVC is vendor-agnostic and supports various types of data storage across all popular cloud vendors, it requires manual intervention to update data stored externally. This is because DVC does not automatically update data storage, which it does not manage to prevent data loss. This limitation hinders the workflow, particularly in rollback scenarios, and is a significant constraint that warrants further improvement.

Additionally, DVC supports versioning at the file level, making it less suitable for environments where file changes are frequent, as this would lead to multiple copies of files in DVC remote, consuming extensive storage. DVC is unsuitable for versioning SQL databases, and exploring other tools is necessary. The suitability of tools for specific use cases is advantageous because overly generalised solutions may not be optimised. However, this necessitates a trade-off between having tools well-suited for specific scenarios and the number of tools that need to be learned.

6.3 Threats to Validity

To ensure the validity of our findings, several measures were taken to address specific threats related to internal validity concerning bias and data accuracy. To minimise personal bias and enhance the transparency of the research, a structured and rigorous approach was implemented for selecting papers and extracting data. All decisions made during the research process were critically evaluated and thoroughly documented to ensure clear and consistent justifications, maintaining objectivity throughout the study. Additionally, tools such as JabRef[1], a reference management software, and Taguette[2], a qualitative data analysis tool, were employed to organise bibliographies, aid in data annotation, and code for enhanced analysis, thereby reducing manual errors and streamlining data management to improve accuracy. Concerning the threat of redundancy, we took specific steps to avoid exaggerating the challenges identified in the study. Each challenge was counted only once per paper, irrespective of the number of times it was mentioned within that paper, and review papers were excluded to avoid counting duplicates from synthesised challenges of other selected studies. During interviews, challenges were documented at the system level rather than per individual practitioner. This approach prevented the over-representation of challenges that might be repeatedly mentioned by multiple individuals within the same system. These measures collectively ensure that our analysis accurately reflects the true prevalence and significance of the challenges, avoiding any potential exaggeration of their frequency or importance. These strategies collectively address internal validity concerns by reducing biases and enhancing the accuracy of the data presented.

Additionally, this study faces a significant threat to external validity due to the limited generalisability of the findings, which stems from conducting the research exclusively within the same multi-organisational company. This limitation is primarily due to the unique operational, cultural, and structural characteristics of the particular company, which may not represent other companies or broader industry contexts. To address this concern, the study deliberately targeted a diverse range of employees from different organisations within the company who are involved in projects related to data pipelines or machine learning. This method not only aids in gathering a broad spectrum of perspectives but also focuses on issues at the project level, which are central to the research objectives. By focusing on specific problems within these projects, the study aims to reduce the external validity threat, ensuring the findings reflect more the dynamics at the project level rather than being skewed by overarching organisational factors.

[1] www.jabref.org
[2] www.taguette.org

7 Conclusion

This study provides a comprehensive analysis of challenges within the academic sphere and industry, highlighting discrepancies in concerns and challenges overlooked in academic studies. It proposes solutions that apply MLOps principles to address selected critical issues, specifically in areas where notable knowledge gaps were identified.

The study revealed that the types of challenges identified through academic literature and professional interviews are largely consistent across both sectors; however, the underlying reasons for these challenges, the factors influencing them, and their prominence in both sectors varied. Issues such as cost, technical guidance, and vendor lock-in, prevalent in the industry, were not discussed in academic settings. Conversely, ethical concerns were notably absent in the industrial context. Another key finding was the heightened emphasis on data source quality and consistency in the interviews. The study thoroughly examined the underlying reasons for these varying levels of concern. These valuable insights will aid in shaping solutions for data pipelines that effectively address real-world problems and address Research Question 1 (RQ1).

In this study, we used these insights to develop requirements and their acceptance criteria for selecting tools that effectively address the challenges of version management and reproducibility in MLOps, especially during the model development phase. This study also includes exploring the problem and solution using a case study synthesised from multiple data science case studies and the insights from practitioners' feedback to refine the solution. This approach ensured that the solutions meet practitioners' needs effectively.

Data Version Control (DVC) emerged as a suitable tool for general scenarios, adeptly meeting the requirements for data tracking and experiment reproducibility while facilitating easy integration. The integration of DVC with Git was validated against user scenarios and practitioners' feedback using the workflows, which underscore the application of MLOps principles such as automation, versioning, and collaboration to address the challenges effectively. The proposed workflows include (1) data tracking with documentation of data version milestones using tags and enabling rollback to different versions, (2) leveraging DVC pipelines for automation and streamlined experiment tracking and reproducibility, and (3) utilising Git to simplify the collaboration of data and experiments along with the code. Moreover, the study discussed the complexities and configurations needed to meet diverse requirements, offering comprehensive insights that aid stakeholders in making informed decisions regarding the tool's applicability and limitations. The proposed solutions and their discussion effectively address Research Question 2 (RQ2).

7.1 Future Work

This study provided insights into various challenges that could be addressed to enhance the design of effective data pipelines and presented solutions to address challenges related to version management. Future work can investigate and address the other challenges to enhance the data pipeline in MLOps. Additionally, future research can explore additional strategies for version management, particularly in scenarios where the current solution is not feasible.

Our data tracking workflow, which leverages version tagging to improve documentation and facilitate data tracking, holds promise for real-world applications. However, its significance in practical settings needs to be evaluated. The development of well-defined semantics for version tags, considering factors such as data complexity, update frequency, compliance, and interoperability, could significantly enhance the documentation and management of data changes. This, in turn, can support data reproducibility and traceability in diverse data science environments.

While the proposed solution, DVC, offers effective data management by storing only the changes between versions without imposing data size limits, it may face performance issues in certain scenarios. For instance, its performance might degrade when dealing with a large number of files or frequent updates to the same files. Future studies could critically examine these limitations and evaluate how DVC's performance scales under different conditions, thereby providing a more comprehensive understanding of its practical applicability.

Furthermore, future research can also apply DVC in complex, real-world machine learning scenarios, such as training custom neural networks. This would offer more profound insights into how DVC and other tools can be integrated and utilised effectively in sophisticated data science and machine learning projects to optimise the development process.

A Referenced Works - Thematic Synthesis

X1: Whang et al., *"Data Collection and Quality Challenges in Deep Learning: A Data-Centric AI Perspective"*

X2: Nahar et al., *"Collaboration Challenges in Building ML-Enabled Systems: Communication, Documentation, Engineering, and Process"*

X3: Weber et al., *"Organizational Capabilities for AI Implementation—Coping with Inscrutability and Data Dependency in AI"*

X4: Mailach and Siegmund, *"Socio-Technical Anti-Patterns in Building ML-Enabled Software: Insights from Leaders on the Forefront"*

X5: Biessmann et al., *"Automated data validation in machine learning systems."*

X6: Campion et al., *"Managing Artificial Intelligence Deployment in the Public Sector"*

X7: De Silva and Alahakoon, *"An artificial intelligence life cycle: From conception to production"*

X8: Chen et al., *"A Comprehensive Study on Challenges in Deploying Deep Learning Based Software"*

X9: Ashmore et al., *"Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges"*

X10: Hukkelberg and Rolland, *"Exploring Machine Learning in a Large Governmental Organization: An Information Infrastructure Perspective"*

X11: Sambasivan et al., *""Everyone wants to do the model work, not the data work": Data Cascades in High-Stakes AI"*

X12: Fischer et al., *"AI System Engineering—Key Challenges and Lessons Learned"*

X13: Jentzsch and Hochgeschwender, *"A qualitative study of Machine Learning practices and engineering challenges in Earth Observation"*

X14: Faubel et al., *"MLOps Challenges in Industry 4.0"*

X15: Muiruri et al., *"Practices and Infrastructures for ML Systems – An Interview Study"*

X16: Rahman et al., *"Machine Learning Application Development: Practitioners' Insights"*

X17: Fredriksson et al., *"Data Labeling: An Empirical Investigation into Industrial Challenges and Mitigation Strategies"*

X18: John et al., *"Developing ML/DL Models: A Design Framework"*

X19: Liu et al., *"Emerging and Changing Tasks in the Development Process for Machine Learning Systems"*

X20: Symeonidis et al., *"MLOps - Definitions, Tools and Challenges"*

X21: Granlund et al., *"MLOps Challenges in Multi-Organization Setup"*

X22: Kreuzberger et al., *"Machine Learning Operations (MLOps): Overview, Definition, and Architecture"*

X23: Lewis et al., *"Software Architecture Challenges for ML Systems"*

X24: Zeydan and Mangues-Bafalluy, *"Recent Advances in Data Engineering for Networking"*

X25: Reddy, *"Data Engineering Challenges in AI automation"*

X26: Côté et al., *"Quality Issues in Machine Learning Software Systems"*

X27: John et al., *"Towards an AI-driven business development framework: A multi-case study"*

X28: Hutchinson et al., *"Towards Accountability for Machine Learning Datasets: Practices from Software Engineering and Infrastructure"*

X29: Haakman et al., *"AI Lifecycle Models Need to Be Revised: An Exploratory Study in Fintech"*

X30: Serban et al., *"Adoption and Effects of Software Engineering Best Practices in Machine Learning"*

X31: Dulac-Arnold et al., *"Challenges of real-world reinforcement learning: definitions, benchmarks and analysis"*

X32: Shukla and Cartlidge, *"Challenges Faced by Industries and Their Potential Solutions in Deploying Machine Learning Applications"*

X33: Barry et al., *"StreamAI: Dealing with Challenges of Continual Learning Systems for Serving AI in Production"*

X34: Nahar et al., *"A Meta-Summary of Challenges in Building Products with ML Components – Collecting Experiences from 4758+ Practitioners"*

X35: Lyu et al., *"An Empirical Study of the Impact of Data Splitting Decisions on the Performance of AIOps Solutions"*

X36: Lwakatare et al., *"DevOps for AI – Challenges in Development of AI-enabled Applications"*

X37: Bodor et al., *"MLOps: Overview of Current State and Future Directions"*

X38: Garg et al., *"On Continuous Integration / Continuous Delivery for Automated Deployment of Machine Learning Models using MLOps"*

X39: Ibáñez and Olmeda, *"Operationalising AI Ethics: How Are Companies Bridging the Gap between Practice and Principles? An Exploratory Study"*

X40: Li et al., *"Testing machine learning systems in industry: an empirical study"*

X41: Paudyal et al., *"Toward Deployments of ML Applications in Optical Networks"*

X42: Eigner et al., *"Towards Resilient Artificial Intelligence: Survey and Research Issues"*

X43: Mäkinen et al., *"Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help?"*

B Interview Questions Template

This appendix presents the questions template used as a reference for the semi-structured interviews, ensuring a comprehensive exploration of various aspects of data management in current projects. These projects span domains such as data engineering, machine learning, and related fields, involving roles such as data engineers, data scientists, data architects, DevOps engineers, product owners, and so on. The questions are designed to cover diverse aspects, including role-specific responsibilities, current projects, data management challenges, tools and technologies used, collaboration practices, and ethical considerations.

1. **Introduction**
    - Can you please briefly introduce me to your role and experience?
    - What specific roles have you been involved in?
2. **Context**
    - What are the current projects that you are working on?
    - How is data typically collected, processed, and stored in your current environment?
    - Do you have a data pipeline? Is it automated?
    - Suppose data is needed from other teams. How do you collaborate with another team to access the data? Have you encountered situations where access to certain data was restricted? If yes, how did this impact the development and execution of machine learning projects?
    - Have you faced any challenges with ensuring effective communication and collaboration?
    - *(For data scientist and data engineer roles)* How do you integrate data from different sources into your machine learning pipelines? Have you faced any compatibility issues, and how were they resolved?
3. **Challenges**
    - What are the major pain points in your current project relating to data?
    - In your experience, what are the common challenges faced with data management in your current project?
    - Can you share a specific example or scenario where you encountered such challenges and how you addressed them?
    - *(For data engineer, data infrastructure engineer, data analyst, DevOps engineer, and data architect roles)* What common challenges are encountered in designing, deploying, or maintaining data pipelines?

- *(For data engineer, data analyst, and data scientist role)* What considerations do you consider when dealing with diverse data formats or structures?
- *(For data engineer or data scientist role)* Do you trust the data you get from different sources?
- *(For data scientist role)* In your experience, how often does data quality become a challenge in machine learning projects, and what are the typical issues you encounter?
- How do you test the data quality?
- *(For machine learning projects)* To what extent do you need domain expertise to understand and work effectively with the data?
- *(For data scientist role)* In your experience, how do you approach the process of data labeling for machine learning projects? Can you share insights into the methods or tools you've found effective and any challenges you've encountered in ensuring accurate and meaningful annotations for training datasets?
- *(For data engineer, data infrastructure engineer, and DevOps engineer roles)* What are some common issues you've encountered in maintaining and troubleshooting data pipelines?
- *(For data engineer, data infrastructure engineer, and DevOps engineer roles)* What strategies or technologies do you typically employ to handle increased workloads and demands on the infrastructure as data and machine learning projects scale?
- *(For data engineer, data infrastructure engineer, and DevOps engineer roles)* What challenges have you encountered in ensuring the scalability and efficiency of data pipelines, especially when dealing with large and diverse data?
- *(For data scientist role)* Have you encountered situations where the data for a machine learning project exhibited biases? If so, how did you become aware of it, and what steps did you take to identify, test, or mitigate bias in the data to ensure fair and unbiased model outcomes?

4. **Tools and technology**
   - What technologies and tools do you commonly use for building and maintaining data pipelines?
   - Have you faced any compatibility or integration challenges with the tools you use for data engineering?
5. **Collaboration**
   - Do you follow specific strategies or best practices when working with cross-functional teams to ensure smooth data collaboration?
6. **Ethical and Legal**
   - Have you faced challenges ensuring data privacy and regulatory compliance in your projects? If yes, how did you handle these situations, and were there specific strategies or frameworks that proved effective?

7. **Reflection**

    - Reflecting on your past projects, is there anything you would have done differently to address challenges or any valuable lessons learned that you can share?

C Description of Case Studies

Project ID	Details
S1	Integration of different data sources such as user behaviour data, third-party data into Snowflake for data processing and reporting, with the inclusion of DBT and Airflow for data model management and historical data adjustments.
S2	Collects user data for permission-based marketing, integrates with email and e-commerce platforms for personalised communication, utilising Databricks, AWS Lambda, S3, RDS, and manages user permissions and data processing.
S3	Tracks and analyses data, uses a redirector to manage and monitor affiliate links with unique IDs, and processes click-out events for detailed analysis. It involves collecting clickstream events and user interactions via Snowplow, processing this data through custom analytics pipelines and employing AWS for robust hosting and data storage solutions.
S4	Enhances data accessibility and processing across the organization, managing data transformations and warehousing with tools like Airflow and Snowflake, supports custom pipeline development, ensures data compliance, and provides a data catalog for transparency and governance. Includes data observability and quality control mechanisms to support business intelligence and analytics needs.
S5	Consolidates data from global subsidiaries using various transfer methods, processed through Airflow and DBT in an ELT pipeline, with Snowflake as the data warehouse. Transitioned from initial external recommended tools setup to a more refined and efficient data integration system.
S6	Enhance user engagement through personalized content delivery in various formats like text, audio, video; leveraging large language models.
S7	Establishes a data pipeline for collecting user demographic data through surveys integrated with tracking data. Utilizes this combined data to train a machine learning model for demographic predictions, with a focus on near real-time data updating and processing for accurate analytics and targeting. It includes data pipeline that download data from Snowflake and continuously trains the machine learning model built using AWS technologies.

Table C.1 continued from previous page

Project ID	Details
S8	Utilises clickstream data to create user behaviour models and predict churn likelihood through machine learning and contrastive learning techniques.
S9	Enhances platform user experience by optimizing features using predictive analytics to improve conversion rates. Besides heuristic front-end enhancements and dashboard analytics, the project leverages Google Analytics, AWS S3, and Athena for data processing.
S10	Develops a search service for finding semantically similar text and images in a digital archive, supporting multi-modal search queries. The backend infrastructure comprises several microservices hosted in a Kubernetes cluster on AWS.
S11	Provides a conversational interface powered by generative AI for exploring content and receiving personalized product recommendations.
S12	Aggregates and processes data from multiple sources for analytics and dashboard creation, supporting comprehensive business intelligence and decision-making. Collects data from different sources to generate comprehensive KPIs, leverages connectors to various data providers and enable SQL-based analysis.
S13	Facilitates the adaptation of multimedia content across different regions,leveraging AI for enhancements. It involves complex data processing and creating a searchable index or vector database for efficient content management and localization.

Table C.1: Case study systems description

Bibliography

[ACP21] R. Ashmore, R. Calinescu, C. Paterson. “Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges”. In: *ACM Computing Surveys* 54.5 (2021), pp. 1–39. ISSN: 1557-7341. DOI: 10.1145/3453444 (cit. on p. 81).

[APLC22] S. Akoush, A. Paleyes, A. V. Looveren, C. Cox. *Desiderata for next generation of ML model serving*. 2022. arXiv: 2210.14665 [cs.LG] (cit. on p. 20).

[AU21] M. A. Al Alamin, G. Uddin. “Quality Assurance Challenges For Machine Learning Software Applications During Software Development Life Cycle Phases”. In: *2021 IEEE International Conference on Autonomous Systems (ICAS)*. 2021, pp. 1–5. DOI: 10.1109/ICAS49788.2021.9551151 (cit. on p. 21).

[BAWX20] U. Bhatt, M. Andrus, A. Weller, A. Xiang. *Machine Learning Explainability for External Stakeholders*. 2020. arXiv: 2007.05408 [cs.CY] (cit. on p. 20).

[BBB23] M. Barry, A. Bifet, J.-L. Billy. “StreamAI: Dealing with Challenges of Continual Learning Systems for Serving AI in Production”. In: *Proceedings of the 45th International Conference on Software Engineering: Software Engineering in Practice*. ICSE-SEIP ’23. Melbourne, Australia: IEEE Press, 2023, pp. 134–137. DOI: 10.1109/ICSE-SEIP58684.2023.00017. URL: https://doi.org/10.1109/ICSE-SEIP58684.2023.00017 (cit. on p. 82).

[BGR21] F. Biessmann, J. Golebiowski, S. P. Rukat T Lange D. “Automated data validation in machine learning systems.” In: *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* (2021) (cit. on p. 81).

[BHN23] A. Bodor, M. Hnida, D. Najima. “MLOps: Overview of Current State and Future Directions”. In: *Lecture Notes in Networks and Systems*. Springer International Publishing, 2023, pp. 156–165. ISBN: 9783031268526. DOI: 10.1007/978-3-031-26852-6_14 (cit. on p. 82).

[BM16] E. Brynjolfsson, K. McElheran. “The rapid adoption of data-driven decision-making”. In: *American Economic Review* 106.5 (2016), pp. 133–139 (cit. on p. 15).

[BWP15] D. Badampudi, C. Wohlin, K. Petersen. “Experiences from using snowballing and database searches in systematic literature studies”. In: *Proceedings of the 19th International Conference on Evaluation and Assessment in Software Engineering* (2015). URL: https://api.semanticscholar.org/CorpusID:15956086 (cit. on pp. 23, 25).

[CBA15] G. G. Claps, R. Berntsson-Svensson, A. Aurum. “On the journey to continuous deployment: Technical and social challenges along the way”. In: *Inf. Softw. Technol.* 57 (2015), pp. 21–31. URL: https://api.semanticscholar.org/CorpusID:205436649 (cit. on p. 18).

[CCL+20] Z. Chen, Y. Cao, Y. Liu, H. Wang, T. Xie, X. Liu. "A Comprehensive Study on Challenges in Deploying Deep Learning Based Software". In: (2020). DOI: 10.48550/ARXIV.2005.00760. arXiv: 2005.00760 [cs.SE] (cit. on pp. 21, 81).

[CD11] D. S. Cruzes, T. Dyba. "Recommended Steps for Thematic Synthesis in Software Engineering". In: *2011 International Symposium on Empirical Software Engineering and Measurement*. 2011, pp. 275–284. DOI: 10.1109/ESEM.2011.36 (cit. on pp. 15, 27).

[CHME20] A. Campion, M.-G. Hernandez, S. Mikhaylov Jankin, M. Esteve. "Managing Artificial Intelligence Deployment in the Public Sector". In: *Computer* 53.10 (2020), pp. 28–37. ISSN: 1558-0814. DOI: 10.1109/mc.2020.2995644 (cit. on p. 81).

[clo] cloud.google. *MLOps: Continuous delivery and automation pipelines in machine learning*. Website. Available online: https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines-in-machine-learning (cit. on pp. 25, 26).

[CNB+23] P.-O. Côté, A. Nikanjam, R. Bouchoucha, I. Basta, M. Abidi, F. Khomh. "Quality Issues in Machine Learning Software Systems". In: (2023). DOI: 10.48550/ARXIV.2306.15007. arXiv: 2306.15007 [cs.SE] (cit. on p. 82).

[DA22] D. De Silva, D. Alahakoon. "An artificial intelligence life cycle: From conception to production". In: *Patterns* 3.6 (June 2022), p. 100489. ISSN: 2666-3899. DOI: 10.1016/j.patter.2022.100489 (cit. on p. 81).

[DLM+21] G. Dulac-Arnold, N. Levine, D. J. Mankowitz, J. Li, C. Paduraru, S. Gowal, T. Hester. "Challenges of real-world reinforcement learning: definitions, benchmarks and analysis". In: *Machine Learning* 110.9 (2021), pp. 2419–2468. ISSN: 1573-0565. DOI: 10.1007/s10994-021-05961-4 (cit. on p. 82).

[DTZ+23] J. Diaz-de-Arcaya, A. I. Torre-Bastida, G. Zárate, R. Miñón, A. Almeida. "A Joint Study of the Challenges, Opportunities, and Roadmap of MLOps and AIOps: A Systematic Survey". In: *ACM Comput. Surv.* 56.4 (Oct. 2023). ISSN: 0360-0300. DOI: 10.1145/3625289. URL: https://doi.org/10.1145/3625289 (cit. on p. 21).

[EAN+17] F. Elberzhager, T. Arif, M. Naab, I. Süß, S. Koban. "From agile development to DevOps: going towards faster releases at high quality – experiences from an industrial context". In: *9th International Conference on Software Quality*. Springer, 2017, pp. 33–44. DOI: 10.1007/978-3-319-49421-0_3 (cit. on p. 18).

[EEK+21] O. Eigner, S. Eresheim, P. Kieseberg, L. D. Klausner, M. Pirker, T. Priebe, S. Tjoa, F. Marulli, F. Mercaldo. "Towards Resilient Artificial Intelligence: Survey and Research Issues". In: *2021 IEEE International Conference on Cyber Security and Resilience (CSR)*. 2021, pp. 536–542. DOI: 10.1109/CSR51186.2021.9527986 (cit. on p. 82).

[EM15] I. El Naqa, M. J. Murphy. *What is machine learning?* Springer, 2015 (cit. on p. 15).

[FEG+20] L. Fischer, L. Ehrlinger, V. Geist, R. Ramler, F. Sobiezky, W. Zellinger, D. Brunner, M. Kumar, B. Moser. "AI System Engineering—Key Challenges and Lessons Learned". In: *Machine Learning and Knowledge Extraction* 3.1 (2020), pp. 56–83. ISSN: 2504-4990. DOI: 10.3390/make3010004 (cit. on p. 81).

[FMBO20] T. Fredriksson, D. I. Mattos, J. Bosch, H. H. Olsson. "Data Labeling: An Empirical Investigation into Industrial Challenges and Mitigation Strategies". In: *Lecture Notes in Computer Science*. Springer International Publishing, 2020, pp. 202–216. ISBN: 9783030641481. DOI: 10.1007/978-3-030-64148-1_13 (cit. on p. 81).

[FS17] B. Fitzgerald, K.-J. Stol. "Continuous software engineering: A roadmap and agenda". In: *Journal of Systems and Software* 123 (Jan. 2017), pp. 176–189 (cit. on p. 18).

[FSE23] L. Faubel, K. Schmid, H. Eichelberger. "MLOps Challenges in Industry 4.0". In: *SN Comput. Sci.* 4.6 (Oct. 2023). DOI: 10.1007/s42979-023-02282-2. URL: https://doi.org/10.1007/s42979-023-02282-2 (cit. on p. 81).

[GKS+21] T. Granlund, A. Kopponen, V. Stirbu, L. Myllyaho, T. Mikkonen. "MLOps Challenges in Multi-Organization Setup". In: *2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN)*. 2021, pp. 82–88. DOI: 10.1109/WAIN52551.2021.00019 (cit. on p. 82).

[GPR+21] S. Garg, P. Pundir, G. Rathee, P. Gupta, S. Garg, S. Ahlawat. "On Continuous Integration / Continuous Delivery for Automated Deployment of Machine Learning Models using MLOps". In: *2021 IEEE Fourth International Conference on Artificial Intelligence and Knowledge Engineering (AIKE)*. challenges related to cont. 2021, pp. 25–28. DOI: 10.1109/AIKE52691.2021.00010 (cit. on p. 82).

[HCHD21] M. Haakman, L. Cruz, H. Huijgens, A. van Deursen. "AI Lifecycle Models Need to Be Revised: An Exploratory Study in Fintech". In: *Empirical Softw. Engg.* 26.5 (Sept. 2021). ISSN: 1382-3256. DOI: 10.1007/s10664-021-09993-1. URL: https://doi.org/10.1007/s10664-021-09993-1 (cit. on p. 82).

[HR20] I. Hukkelberg, K. Rolland. "Exploring Machine Learning in a Large Governmental Organization: An Information Infrastructure Perspective". In: *ECIS 2020 Research-in-Progress Papers*. 2020 (cit. on p. 81).

[HSH+21] B. Hutchinson, A. Smart, A. Hanna, E. Denton, C. Greer, O. Kjartansson, P. Barnes, M. Mitchell. "Towards Accountability for Machine Learning Datasets: Practices from Software Engineering and Infrastructure". In: *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*. FAccT '21. ACM, 2021. DOI: 10.1145/3442188.3445918 (cit. on p. 82).

[IO22] J. C. Ibáñez, M. V. Olmeda. "Operationalising AI Ethics: How Are Companies Bridging the Gap between Practice and Principles? An Exploratory Study". In: *AI Soc.* 37.4 (Dec. 2022), pp. 1663–1687. ISSN: 0951-5666. DOI: 10.1007/s00146-021-01267-0. URL: https://doi.org/10.1007/s00146-021-01267-0 (cit. on p. 82).

[JH21] S. Jentzsch, N. Hochgeschwender. "A qualitative study of Machine Learning practices and engineering challenges in Earth Observation". In: *it - Information Technology* 63.4 (2021), pp. 235–247. ISSN: 1611-2776. DOI: 10.1515/itit-2020-0045 (cit. on p. 81).

[JOB20] M. M. John, H. H. Olsson, J. Bosch. "Developing ML/DL Models: A Design Framework". In: *Proceedings of the International Conference on Software and System Processes*. ICSSP '20. ACM, 2020. DOI: 10.1145/3379177.3388892 (cit. on p. 81).

[JOB21] M. M. John, H. H. Olsson, J. Bosch. "Towards MLOps: A Framework and Maturity Model". In: *2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. 2021, pp. 1–8. DOI: 10.1109/SEAA53835.2021.00050 (cit. on pp. 20, 25).

[JOB22] M. M. John, H. H. Olsson, J. Bosch. "Towards an AI-driven business development framework: A multi-case study". In: *Journal of Software: Evolution and Process* 35.6 (Feb. 2022). ISSN: 2047-7481. DOI: 10.1002/smr.2432 (cit. on p. 82).

[KC07] B. Kitchenham, S. Charters. *Guidelines for performing Systematic Literature reviews in Software Engineering*. 2007 (cit. on p. 23).

[KKH23] D. Kreuzberger, N. Kühl, S. Hirschl. "Machine Learning Operations (MLOps): Overview, Definition, and Architecture". In: *IEEE Access* 11 (2023), pp. 31866–31879. DOI: 10.1109/ACCESS.2023.3262138 (cit. on pp. 18–20, 82).

[KL23] A. B. Kolltveit, J. Li. "Operationalizing machine learning models: a systematic literature review". In: *Proceedings of the 1st Workshop on Software Engineering for Responsible AI*. SE4RAI '22. Pittsburgh, Pennsylvania: Association for Computing Machinery, 2023, pp. 1–8. ISBN: 9781450393195. DOI: 10.1145/3526073.3527584. URL: https://doi.org/10.1145/3526073.3527584 (cit. on p. 21).

[KPYL08] S. Kim, S. Park, J. Yun, Y. Lee. "Automated Continuous Integration of Component-Based Software: An Industrial Experience". In: *2008 23rd IEEE/ACM International Conference on Automated Software Engineering*. 2008, pp. 423–426. DOI: 10.1109/ASE.2008.64 (cit. on p. 18).

[LCB20] L. E. Lwakatare, I. Crnkovic, J. Bosch. "DevOps for AI – Challenges in Development of AI-enabled Applications". In: *2020 International Conference on Software, Telecommunications and Computer Networks (SoftCOM)*. IEEE, 2020. DOI: 10.23919/softcom50211.2020.9238323 (cit. on p. 82).

[LERH20] H. Liu, S. Eksmo, J. Risberg, R. Hebig. "Emerging and Changing Tasks in the Development Process for Machine Learning Systems". In: *Proceedings of the International Conference on Software and System Processes*. ICSSP '20. ACM, 2020. DOI: 10.1145/3379177.3388905 (cit. on p. 81).

[LGL+22] S. Li, J. Guo, J.-G. Lou, M. Fan, T. Liu, D. Zhang. "Testing machine learning systems in industry: an empirical study". In: *Proceedings of the 44th International Conference on Software Engineering: Software Engineering in Practice*. ICSE '22. ACM, May 2022. DOI: 10.1145/3510457.3513036 (cit. on p. 82).

[LLS+21] Y. Lyu, H. Li, M. Sayagh, Z. M. Jiang, A. E. Hassan. "An Empirical Study of the Impact of Data Splitting Decisions on the Performance of AIOps Solutions". In: *ACM Transactions on Software Engineering and Methodology* 30.4 (2021), pp. 1–38. ISSN: 1557-7392. DOI: 10.1145/3447876 (cit. on p. 82).

[LOX21] G. A. Lewis, I. Ozkaya, X. Xu. "Software Architecture Challenges for ML Systems". In: *2021 IEEE International Conference on Software Maintenance and Evolution (ICSME)*. 2021, pp. 634–638. DOI: 10.1109/ICSME52107.2021.00071 (cit. on p. 82).

[MG22] B. M. A. Matsui, D. H. Goya. "MLOps: five steps to guide its effective implementation". In: *Proceedings of the 1st International Conference on AI Engineering: Software Engineering for AI*. CAIN '22. Pittsburgh, Pennsylvania: Association for Computing Machinery, 2022, pp. 33–34. ISBN: 9781450392754. DOI: 10.1145/3522664.3528611. URL: https://doi.org/10.1145/3522664.3528611 (cit. on p. 22).

[MLKM21] D. Muiruri, L. E. Lwakatare, J. K. Nurminen, T. Mikkonen. "Practices and Infrastructures for ML Systems – An Interview Study". In: (2021). DOI: 10.36227/techrxiv.16939192.v1 (cit. on pp. 21, 81).

[MLO] MLOps.org. *Machine learning operations*. Website. Available online: https://mlops.org/ (cit. on pp. 18–20).

[MLWS22] S. Mei, C. Liu, Q. Wang, H. Su. "Model Provenance Management in MLOps Pipeline". In: *Proceedings of the 2022 8th International Conference on Computing and Data Engineering*. ICCDE '22. Bangkok, Thailand: Association for Computing Machinery, 2022, pp. 45–50. ISBN: 9781450395717. DOI: 10.1145/3512850.3512861. URL: https://doi.org/10.1145/3512850.3512861 (cit. on p. 19).

[MS23] A. Mailach, N. Siegmund. "Socio-Technical Anti-Patterns in Building ML-Enabled Software: Insights from Leaders on the Forefront". In: *Proceedings of the 45th International Conference on Software Engineering*. ICSE '23. Melbourne, Victoria, Australia: IEEE Press, 2023, pp. 690–702. ISBN: 9781665457019. DOI: 10.1109/ICSE48619.2023.00067. URL: https://doi.org/10.1109/ICSE48619.2023.00067 (cit. on p. 81).

[MSLM21] S. Mäkinen, H. Skogström, E. Laaksonen, T. Mikkonen. "Who Needs MLOps: What Data Scientists Seek to Accomplish and How Can MLOps Help?" In: *2021 IEEE/ACM 1st Workshop on AI Engineering - Software Engineering for AI (WAIN)*. 2021, pp. 109–112. DOI: 10.1109/WAIN52551.2021.00024 (cit. on p. 82).

[MW15] N. Marz, J. Warren. *Big Data: Principles and Best Practices of Scalable Real-time Data Systems*. New York: Manning Publications Co., 2015 (cit. on p. 19).

[NKD13] V. I. Nnebedum, A. U. Kamalu, J. N. Dike. "From Data Management to Data Engineering". In: *International Journal of Computer Applications* 81.11 (Nov. 2013), pp. 36–39. ISSN: 0975-8887. DOI: 10.5120/14059-2300. URL: https://ijcaonline.org/archives/volume81/number11/14059-2300/ (cit. on p. 18).

[NZL+23] N. Nahar, H. Zhang, G. Lewis, S. Zhou, C. Kästner. "A Meta-Summary of Challenges in Building Products with ML Components – Collecting Experiences from 4758+ Practitioners". In: *2023 IEEE/ACM 2nd International Conference on AI Engineering – Software Engineering for AI (CAIN)*. 2023, pp. 171–183. DOI: 10.1109/CAIN58948.2023.00034 (cit. on p. 82).

[NZLK22] N. Nahar, S. Zhou, G. Lewis, C. Kästner. "Collaboration Challenges in Building ML-Enabled Systems: Communication, Documentation, Engineering, and Process". In: *Proceedings of the 44th International Conference on Software Engineering*. ICSE '22. Pittsburgh, Pennsylvania: Association for Computing Machinery, 2022, pp. 413–425. ISBN: 9781450392211. DOI: 10.1145/3510003.3510209. URL: https://doi.org/10.1145/3510003.3510209 (cit. on pp. 21, 81).

[PF13] F. Provost, T. Fawcett. "Data science and its relationship to big data and data-driven decision making". In: *Big data* 1.1 (2013), pp. 51–59 (cit. on p. 15).

[PSYS21] P. Paudyal, S. Shen, S. Yan, D. Simeonidou. “Toward Deployments of ML Applications in Optical Networks”. In: *IEEE Photonics Technology Letters* 33.11 (2021), pp. 537–540. ISSN: 1941-0174. DOI: 10.1109/lpt.2021.3074586 (cit. on p. 82).

[PTRC07] K. Peffers, T. Tuunanen, M. A. Rothenberger, S. Chatterjee. “A Design Science Research Methodology for Information Systems Research”. In: *Journal of Management Information Systems* 24.3 (2007) (cit. on p. 28).

[Pul24] M. R. Pulicharla. “Data Versioning and Its Impact on Machine Learning Models”. In: *Journal of Science & Technology* 5.1 (2024), pp. 22–37 (cit. on p. 20).

[RBOW20] A. Raj, J. Bosch, H. H. Olsson, T. J. Wang. “Modelling Data Pipelines”. In: *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. 2020, pp. 13–20 (cit. on pp. 19, 20).

[RDW+21] V. J. Reddi, G. Diamos, P. Warden, P. Mattson, D. Kanter. *Data Engineering for Everyone*. 2021. arXiv: 2102.11447 [cs.LG] (cit. on pp. 15, 17).

[Red23] D. Reddy. “Data Engineering Challenges in AI automation”. In: *2023 International Conference on Computing, Electronics & Communications Engineering (iCCECE)*. IEEE, 2023. DOI: 10.1109/iccece59400.2023.10238496 (cit. on p. 82).

[RKH+21] M. S. Rahman, F. Khomh, A. Hamidi, J. Cheng, G. Antoniol, H. Washizaki. “Machine Learning Application Development: Practitioners’ Insights”. In: *ArXiv* abs/2112.15277 (2021). URL: https://api.semanticscholar.org/CorpusID:245634201 (cit. on p. 81).

[SBHV20] A. Serban, K. van der Blom, H. Hoos, J. Visser. “Adoption and Effects of Software Engineering Best Practices in Machine Learning”. In: *Proceedings of the 14th ACM / IEEE International Symposium on Empirical Software Engineering and Measurement (ESEM)*. ESEM ’20. ACM, 2020. DOI: 10.1145/3382494.3410681 (cit. on p. 82).

[SC22] R. M. Shukla, J. Cartlidge. “Challenges Faced by Industries and Their Potential Solutions in Deploying Machine Learning Applications”. In: *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*. IEEE, 2022, pp. 0119–0124. DOI: 10.1109/CCWC54503.2022.9720900 (cit. on p. 82).

[SFR23] M. Steidl, M. Felderer, R. Ramler. “The pipeline for the continuous development of artificial intelligence models—Current state of research and practice”. In: *J. Syst. Softw.* 199.C (2023). ISSN: 0164-1212. DOI: 10.1016/j.jss.2023.111615. URL: https://doi.org/10.1016/j.jss.2023.111615 (cit. on p. 21).

[SKH+21] N. Sambasivan, S. Kapania, H. Highfill, D. Akrong, P. Paritosh, L. M. Aroyo. ““Everyone wants to do the model work, not the data work”: Data Cascades in High-Stakes AI”. In: *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*. CHI ’21. ACM, 2021. DOI: 10.1145/3411764.3445518 (cit. on p. 81).

[SKS21] K. Salama, J. Kazmierczak, D. Schut. “Practitioners guide to MLOps: A framework for continuous delivery and automation of machine learning.” In: *Google Cloud: White Paper* (2021). URL: https://services.google.com/fh/files/misc/practitioners_guide_to_mlops_whitepaper.pdf (cit. on pp. 18, 20).

[SM22] K. Shivashankar, A. Martini. “Maintainability Challenges in ML: A Systematic Literature Review”. In: *2022 48th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*. 2022, pp. 60–67. DOI: 10.1109/SEAA56994.2022.00018 (cit. on p. 21).

[SNKP22] G. Symeonidis, E. Nerantzis, A. Kazakis, G. A. Papakostas. “MLOps - Definitions, Tools and Challenges”. In: *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*. 2022, pp. 0453–0460. DOI: 10.1109/CCWC54503.2022.9720902 (cit. on p. 82).

[SV22] A. Serban, J. Visser. “Adapting Software Architectures to Machine Learning Challenges”. In: *2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER)*. Los Alamitos, CA, USA: IEEE Computer Society, 2022, pp. 152–163. DOI: 10.1109/SANER53432.2022.00029. URL: https://doi.ieeecomputersociety.org/10.1109/SANER53432.2022.00029 (cit. on p. 22).

[TBF+22] M. Testi, M. Ballabio, E. Frontoni, G. Iannello, S. Moccia, P. Soda, G. Vessio. “MLOps: A Taxonomy and a Methodology”. In: *IEEE Access* 10 (2022), pp. 63606–63618. DOI: 10.1109/ACCESS.2022.3181730 (cit. on pp. 17–20).

[WES+22] M. Weber, M. Engert, N. Schaffer, J. Weking, H. Krcmar. “Organizational Capabilities for AI Implementation—Coping with Inscrutability and Data Dependency in AI”. In: *Information Systems Frontiers* 25.4 (June 2022), pp. 1549–1569. ISSN: 1387-3326. DOI: 10.1007/s10796-022-10297-y. URL: https://doi.org/10.1007/s10796-022-10297-y (cit. on p. 81).

[WKRM22] C. Wohlin, M. Kalinowski, K. Romero Felizardo, E. Mendes. “Successful combination of database search and snowballing for identification of primary studies in systematic literature studies”. In: *Information and Software Technology* 147 (July 2022), p. 106908. ISSN: 0950-5849. DOI: 10.1016/j.infsof.2022.106908. URL: http://dx.doi.org/10.1016/j.infsof.2022.106908 (cit. on p. 15).

[Woh14] C. Wohlin. “Guidelines for snowballing in systematic literature studies and a replication in software engineering”. In: (2014). DOI: 10.1145/2601248.2601268. URL: https://doi.org/10.1145/2601248.2601268 (cit. on p. 23).

[WRSL23] S. E. Whang, Y. Roh, H. Song, J.-G. Lee. “Data Collection and Quality Challenges in Deep Learning: A Data-Centric AI Perspective”. In: *The VLDB Journal* 32.4 (Jan. 2023), pp. 791–813. ISSN: 1066-8888. DOI: 10.1007/s00778-022-00775-9. URL: https://doi.org/10.1007/s00778-022-00775-9 (cit. on p. 81).

[ZB13] H. Zhang, M. A. Babar. “Systematic reviews in software engineering: An empirical investigation”. In: *Information and software technology* 55.7 (2013), pp. 1341–1354 (cit. on p. 23).

[ZM22] E. Zeydan, J. Mangues-Bafalluy. “Recent Advances in Data Engineering for Networking”. In: *IEEE Access* 10 (2022), pp. 34449–34496. ISSN: 2169-3536. DOI: 10.1109/ACCESS.2022.3162863 (cit. on pp. 18, 82).

[ZYD20] Y. Zhou, Y. Yu, B. Ding. “Towards MLOps: A Case Study of ML Pipeline Platform”. In: *2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*. 2020, pp. 494–500. DOI: 10.1109/ICAICE51518.2020.00102 (cit. on p. 22).