CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden
www.chalmers.se

LICENTIATE THESIS

Data is considered a valuable resource by companies as it can be used to make more-informed business decisions, improve marketing campaigns, optimize business operations and reduce costs, all with the goal of increasing revenue and profits. However, with the increased importance of data, existing data management strategies became obsolete and futile. Data silos, data set inconsistencies and many other data quality problems led to faulty and worthless findings. This research attempts to improve the data management approach at the industrial level. Based on an exploratory multiple case study, a series of data management challenges and considerations were identified for applying deep learning. A multi-vocal literature review and multiple case studies were employed to analyze the state-of-the-art data management approaches. In light of the ongoing trend of Artificial Intelligence and the importance of data management, this study tried to model a robust data pipeline for developing AI-enhanced embedded systems. The research also contributes to identifying the potential challenges while building and maintaining data pipelines. Equally important, the research provides a closer look at the faults at the various stages of a data pipeline and corresponding mitigation strategies. The licentiate thesis is intended for both academic and industry readers. Researchers can pay attention to the practical data management challenges that are not addressed in this thesis. Practitioners from the industry can reflect on the role and importance of adopting appropriate data management practices when developing and using AI-enhanced systems in the context of the embedded system companies.

CHALMERS
UNIVERSITY OF TECHNOLOGY

AISWARYA RAJ MUNAPPY • Data management and Data Pipelines:An empirical investigation in the embedded systems domain • 2021

Data management and Data Pipelines: An empirical investigation in the embedded systems domain

AISWARYA RAJ MUNAPPY

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2021
www.chalmers.se

THESIS FOR THE DEGREE OF LICENTIATE OF ENGINEERING

Data management and Data Pipelines: An empirical investigation in the embedded systems domain

Aiswarya Raj Munappy

Division of Software Engineering
Department of Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden, 2021

Data management and Data Pipelines:
An empirical investigation in the embedded systems domain

Aiswarya Raj Munappy

ISSN 1652-876X

Department of Computer Science and Engineering
Chalmers University of Technology
SE-412 96 Gothenburg, Sweden
Phone: +46 (0)31 772 1000
www.chalmers.se

This thesis has been prepared using LaTeX.
Printed by Chalmers Reproservice
Gothenburg, Sweden, May2021

Dedicated to my husband Sarath

Abstract

Context: Companies are increasingly collecting data from all possible sources to extract insights that help in data-driven decision-making. Increased data volume, variety, and velocity and the impact of poor quality data on the development of data products are leading companies to look for an improved data management approach that can accelerate the development of high-quality data products. Further, AI is being applied in a growing number of fields and thus it is evolving as a horizontal technology. Consequently, AI components are increasingly been integrated into embedded systems along with electronics and software. We refer to these systems as AI-enhanced embedded systems. Given the strong dependence of AI on data, this expansion also creates a new space for applying data management techniques.
Objective: The overall goal of this thesis is to empirically identify the data management challenges encountered during the development and maintenance of AI-enhanced embedded systems, propose an improved data management approach and empirically validate the proposed approach.
Method: To achieve the goal, we conducted this research in close collaboration with Software Center companies using a combination of different empirical research methods: case studies, literature reviews, and action research.
Results and conclusions: This research provides five main results. First, it identifies key data management challenges specific to Deep Learning models developed at embedded system companies. Second, it examines the practices such as DataOps and data pipelines that help to address data management challenges. We observed that DataOps is the best data management practice that improves the data quality and reduces the time to develop data products. The data pipeline is the critical component of DataOps that manages the data life cycle activities. The study also provides the potential faults at each step of the data pipeline and the corresponding mitigation strategies. Finally, the data pipeline model is realized in a small piece of data pipeline and calculated the percentage of saved data dumps through the implementation.
Future work: As future work, we plan to realize the conceptual data pipeline model so that companies can build customized robust data pipelines. We also plan to analyze the impact and value of data pipelines in cross-domain AI systems and data applications. We also plan to develop AI-based fault detection and mitigation system suitable for data pipelines.

List of Publications

This thesis is based on the following publications:

[A] **Munappy, A. R.**, Bosch, J., Olsson, H. H., Arpteg, A., Brinne, B, "Data Management Challenges for Deep Learning". *45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pp. 140-147, 2020. IEEE..

[B] **Munappy, A. R.**, Mattos, D. I., Bosch, J., Olsson, H. H., Dakkak, A., "From Ad-Hoc Data Analytics to DataOps". *In Proceedings of the International Conference on Software and System Processes*, (pp. 165-174), (2020, June), Association for Computing Machinery.

[C] **Munappy A. R.**, Bosch, J., Olsson, H. H., Wang, T. J., "Modelling Data Pipelines". In 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 13-20). IEEE..

[D] **Munappy, A. R.**, Bosch, J., Olsson, H. H., "Data Pipeline Management in Practice: Challenges and Opportunities ". *In International Conference on Product-Focused Software Process Improvement (pp. 168-184). Springer, Cham..*

[E] **Munappy A. R.**, Bosch, J., Olsson, H. H., Wang, T. J., "Towards Automated Detection of Data Pipeline Faults". *27th Asia-Pacific Software Engineering Conference (APSEC) (pp. 346-355). IEEE..*

Other publications by the author, not included in this thesis, are:

[F] LE. Lwakatare, **A. R. Munappy**, J. Bosch, HH. Olsson, I. Crnkovic, "A taxonomy of software engineering challenges for machine learning systems: An empirical investigation". *Proc. In International Conference on Agile Software Development*, Springer, Cham, 2019.

[G] Large-scale machine learning systems in real-world industrial settings: A review of challenges and solutions., "LE. Lwakatare, **A. R. Munappy**, I. Crnkovic, J. Bosch, HH. Olsson". *Information and Software Technology 127 (2020): 106368.*

Personal Contribution

For all publications where I am the first author, my contribution is listed below using the CRediT (Contributor Roles Taxonomy) author statement [1], where I made the following contributions:

1. Conceptualization – Ideas; formulation of overarching research goals and aims
2. Methodology – Development or design of methodology; creation of models
3. Validation - Focus on the overall replication/reproducibility of results
4. Investigation - Conducting a research and performing data collection.
5. Data Curation – Activities to annotate scrub data and maintain research data
6. Writing – Original draft, review, and editing
7. Preparation – Creation and/or presentation of the published work
8. Project Administration – Management and coordination responsibility for research activities.

For the mentioned papers F and G, in which I am listed as a second author, I made the following contributions:

1. Conceptualization – Formulation of overarching research goals and aims
2. Methodology – Design of methodology and creation of research models.
3. Investigation (partly) – Conducting the research process and performing data collection
4. Data Curation – Activities to annotate scrub data and maintain research data
5. Writing (partly) – Original draft, review, and editing.
6. Project Administration – Management and coordination responsibility for research activities

Acknowledgments

First, I would like to express my sincere gratitude to my main supervisor Jan Bosch for the continuous support of my Ph.D. study and related research, for his patience, support, motivation, and immense knowledge. His guidance helped me throughout the research especially his insightful feedback and advice pushed me to sharpen my thinking and brought my work to a higher level. I could not have imagined having a better advisor and mentor for my Ph.D. study. Second, I would like to thank my co-advisor Helena Holmström Olsson. Although she contributed to my research in many different ways, I would like to focus on two in particular. First, her expertise was invaluable in formulating the research questions and methodology. Second, her insights and tips on conducting interviews, how to analyze the data, and her continuous availability has helped to build a strong foundation for my qualitative research. I would like to thank everyone at Chalmers University who has been involved in the process of this research, especially my examiner Ivica Crnkovic. In addition to him, I wish to express my gratitude to Eric Knauss, who was responsible to plan my teaching activities and made this process smooth, Lucy Ellen Lwakatare, whom I also learned a lot from during our common lunches and breaks, and finally Anders Jansson and Tian J. Wang, who provided input to parts of my research and made the research work and experience exciting out of the office. Last but not least, a great thanks to Software Center companies with which I had the opportunity to collaborate more closely during this research. Without them, this research would not be possible. The practitioners in the companies were always cooperative and encouraging.

Contents

CHAPTER 1

Introduction

Data is a revolution that will transform our life, work, and even thoughts [1]. The amount of data is exploding at an extraordinary pace due to the developments in mobile and sensing devices, social media, and web technologies. For instance, Twitter processes over 70M tweets per day, thereby producing over 8TB daily [2]. Google generates around 2.5 million Terabytes per day. Google predicts that about 175 zettabytes of data will be generated worldwide by 2025 [3]. Over 90% of the world's data was generated during the last couple of years[4]. Products whose primary objective is to use data to facilitate an end goal are referred to as data products. For instance, Google Analytics is a data product as its primary objective is bringing a quantitative understanding of online behavior to the user. On the contrary, Instagram is not a data product by definition while the functionalities in Instagram like tagging and searching are data products. Data products can be broadly classified as raw data, derived data, algorithms, decision support, and automated decision-making [5]. Raw data is simply collected from the source and stored for future use without further processing whereas derived data is processed raw data. Algorithms consume data and return insights. Google Image is an example that accepts an input image from the user and outputs similar

images. Behind the scenes, the product extracts features, classifies the image, matches it to stored images by calculating a similarity index, and returns similar images. Decision support systems such as Google Analytics provide information and thus offer help with decision-making. Design decisions in data collection and derivation of new data will be done by the decision support system. Nevertheless, the interpretation of results is made by the users. In automated decision-making systems, the algorithm does all the work with the data and algorithm and presents the user with the final output. Netflix movie recommendations is an example of an automated decision-making system [5]. Companies are increasingly using the data they collected over the years for different purposes. Data is used for quality assurance and diagnostics so that troubleshooting efforts can be significantly reduced. Some companies use data for features, functionality improvement, and performance optimization. Several companies consider data as an asset to create useful and comparative insights drawn from it [6]. Thus, data can be considered as a driver of innovation, and the cornerstone to attaining competitive advantage to the business [7]. On the other hand with the increase in volume, variety, velocity, and application of data, data management is becoming increasingly important as well as challenging.

1.1 Data and AI Applications

Organizations are increasingly adopting AI to glean knowledge from data and implement a diverse set of computationally hard tasks, ranging from machine perception to text understanding, health care, genomics, and even the protection of endangered species [8]. Large scale online companies like Alphabet (Google), Apple, Facebook, Microsoft, Amazon, etc. are investing heavily in Artificial Intelligence. Nevertheless, empirical studies [9] and experience reports [10] [11] [12] [13] published across different disciplines present the challenges encountered by operational AI applications. The performance and quality of AI models are very much dependent on the data fed to them. In the studies by Google [14] and Microsoft [15] analyzing the steps of AI model development, steps related to data and data quality management are more compared to others. Moreover, creating a dataset is the first step in the AI process. Errors in the first step will be propagated through the remaining steps resulting in the creation of poor-performing models. Moreover, the role of data

management is evident from the success of large scale companies like Apple (with its intelligent Siri) [16], Amazon (with its ever-improving Alexa) [17], Facebook and Google (with their image recognition algorithms) [18]. All these companies deal with ample amounts of speech, voice, and image data. These companies have access to large sets of data and their applications rely on that data to make quick and smart decisions. Thus, companies dealing with large sets of data and knows how to organize and manage data can yield more benefits compared to others. In this context, it is clear that efficient collection and management of data is the main challenge for software companies to be on par with their competitors.

1.2 Need for Data Management

Data management is necessary to increase productivity, reduce data loss and improve data quality. Well-organized and streamlined data enables accessibility and availability of data for the teams working with it. Data quality is paramount as dirty data is not very useful for the business. Data management practices that automatically clean and organize the data are a method to ensure data quality. Data loss is very common and without data management practices, it will remain unidentified and finally leads to the development of bad quality data products [19]. Data management is the organization of data, the steps used to achieve efficiency, and gather intelligence from that data [20]. For many years data management has been considered an important step and it was used to automate traditional information processing in the early days [21]. Later, data management allowed fast, reliable, and secure access to globally distributed data. Data warehousing has a significant contribution in formalizing data architecture and data management practices. Resolving inconsistencies among redundant data sources, providing an understandable source of data for business user access, reducing the complexity of tangled and fragile point-to-point application interfaces [20] were the major attractions of data warehouses. Data warehouses were deployed with Relational Database Management System(RDBMS) technology and the applications were largely limited to Business Intelligence and reporting [22]. Raw data products, as well as derived data products, still use data warehouses. Data lakes were the next big shift in data management due to the advances in data management and demands for a new approach. The data lake became dominant database

technology as it could overcome the limitation of RDBMS which is to impose schema before storing data in a database [23]. Data lakes are used as data storage for decision-supporting systems as well as for automated decision-making systems. However, they demand multiple levels of data refinement, ranging from raw data for data scientists to integrated and aggregated data for basic reporting, and thus need new best practices along with data lakes [23]. Artificial Intelligence (AI) models are the typical example for decision-supporting systems or automated decision-making systems.

1.3 Data Management for AI Applications

Although many of the medium and small-scale companies are trying to adopt AI, challenges in the development phases are holding them back from yielding the full potential. The development of AI applications in real-time settings is non-trivial and the development process differs from that of traditional software engineering [24]. Currently, there is a growing interest and need to understand how AI applications are developed, deployed, and maintained over time in real-world commercial settings. As stated by Lwakatare et. al, the development stages of AI models are classified into 4 major stages namely data acquisition, model creation, training/evaluation, and deployment [25]. In the data acquisition step, data for training, validation, and testing are gathered and a dataset is created. When creating AI models, typically several experiments are conducted before selecting the final AI model. During AI model creation, the data is given as input to different learning algorithms in a trial and error fashion, and the performance of AI models is evaluated using validation data (a part of training data). Most studies in academia are tended to focus on theoretical breakthroughs of learning algorithms for AI. However, empirical studies show that they constitute only a small part of the operational ML systems [14]. The AI engineering challenges identified by Lwakatare et al [25], Arpteg et. al [26] and Amershi et. al [15] clearly shows that challenges related to data and data management are more compared to other challenges. Besides, data being the backbone of AI models, data errors can cause severe performance degradation. Although data and data management has a long history and has been discussed by scientists, statisticians, librarians, computer scientists, and others for years AI models require special kinds of data management strategies. Because development and maintenance

of AI models raise unique data challenges such as metadata management, data shifts, class imbalance and so on [27]. From the papers that discuss the challenges of operational AI applications, it can be easily observed that the challenges related to data management are significant [28].

1.4 Limitations of existing Data Management Practices

An increasing number of studies have found that data management is a challenge faced by AI practitioners [27] [28] [15]. However, the aforementioned studies have tended to focus on challenges rather than solutions which raises many questions on how to efficiently manage the data in real-world commercial settings. Our research seeks to improve the existing data management approach such that it can be used for building and maintaining high-quality data products in particular AI-enhanced embedded systems i.e., systems that involve both "traditional" software and Artificial Intelligence components. In AI-enhanced embedded systems, AI components are used to enhance the existing functionalities. For example, Analytic dashboards that convert data into insights. AI components here can be replaced by rule-based components or statistical models. Further, incorporating AI components is comparatively difficult in embedded systems. Currently, the embedded systems industry is in significant transition, i.e. markets are more fast-changing and unpredictable, customer requirements becoming increasingly complex and rapidly advancing technologies [29]. While the ability to manufacture high-quality mechanical systems remains critical, embedded systems companies are employing AI-components/data products together with electronics and software. This requires a significant improvement in their data management practices, and currently many large companies within the embedded systems domain struggle with adopting the right practice. Previous work has focused on incorporating AI in web-based systems like recommender systems, AI-intensive systems like weather prediction systems, and AI-powered systems like self-driving cars. Furthermore, this thesis aims to validate the improved data management approach by evaluating the impact of the changes introduced. We use DataOps as a data management approach to solve data management challenges. Data pipelines being the critical component in DataOps, we have researched modeling robust data pipelines as well. To model a robust data

pipeline, two functionalities namely fault detection and mitigation strategies are required. Therefore, we have done studies on fault detection and mitigation strategies adopted by various embedded system companies. In the literature, there are a few examples of data pipelines. However, no work mentions the data pipelines specifically modeled for DataOps and AI-enhanced embeddedd systems.

1.5 Contributions of the Research

The contributions of this thesis are manifold. First, it provides an overview of data management challenges for industrial deep learning models. Second, it investigates how the embedded system company adapts to the increasing significance of data and what are the changes in data management practices over time. Third, it identifies the opportunities as well as challenges encountered by the industries during the development and maintenance of data pipelines. Forth, it develops a conceptual model for data pipelines that is suitable for the embedded system companies using data products such as Google Analytics. Fifth, it investigates the potential faults at each step of the data pipeline and the corresponding mitigation strategies.

1.6 Structure of Chapters

This thesis is organized as follows. Chapter 1 introduces the topic of data management for AI as well as presenting the goals of this Ph.D. research, the research questions, and the key contributions. Chapter 2 presents the background of the study. Chapter 3 discusses the research methodology, as well as the research questions and the motivation for using each of the methods in the studies. Chapters 4, 5, 6, 7, and 8 are based on the publications A to E and constitute the key contributions of this thesis. Chapter 4 discusses data management challenges for industrial deep learning systems. Chapter 5 describes the definition for DataOps and the DataOps evolution stages. Chapter 6 details the opportunities and challenges with the data pipeline development and maintenance. Chapter 7 proposes a new conceptual data pipeline model for the data management approach proposed in Chapter 5. Chapter 8 discusses the development of robust data pipelines. Chapter 9 concludes the thesis with a discussion of the main results and future work.

CHAPTER 2

Background

This thesis studies data management and data pipelines, specifically for AI-enhanced systems, and, in order to provide the reader with the necessary information needed to better understand the remainder of the thesis, this section provides background information and describes the related work of this thesis. Section 2.1 discusses the different applications of data from which the reader can understand the influence of data in industries. Further, it defines data products and the categorization of data products which is a prerequisite for understanding the data pipeline model discussed in chapter 7 and chapter 8. 3Vs of big data is discussed in section 2.2 which helps in understand why the study of data management challenges is essential. Section 2.3, presents the synergy between Big Data and AI together with examples, such as opportunities with big data, applications of AI, etc. Section 2.3 is the basis for all the chapters presented in this thesis. Section 2.4 presents the concept of data management and why is it required to have an improved data management approach which can be used to understand chapters 4 and 5. Section 2.5 explains the AI-embedded systems and the need for data management in such systems which help reader to familiarize the concepts further explained in chapter 8. Section 2.6 and 2.7 gives a brief overview of Agile methodol-

ogy and DevOps respectively as DataOps borrows most heavily from DevOps, Agile, and statistical process control. The conceptual model of data pipelines is an important contribution of this thesis. To enable a better understanding of the modeling of data pipelines, challenges, and opportunities, section 2.9 presents an outline of data pipelines and the need for automated data pipelines. Finally, section 2.10 summarizes the chapter

2.1 Data Products and Applications

A data product is an application or tool that uses data to help businesses improve decisions and processes. Data products provide a friendly user interface applying data science to provide predictive analytics, descriptive data modeling, data mining, machine learning, risk management, and a variety of analysis methods to non-data scientists [30]. Data products can be broadly classified as raw data, derived data, algorithms, decision support, and automated decision-making [5]. Raw data is simply collected from the source and stored for future use without further processing whereas derived data is processed raw data. Algorithms consume data and return insights. Google Image is an example that accepts an input image from the user and outputs similar images. Behind the scenes, the product extracts features, classifies the image, matches it to stored images by calculating similarity index, and returns similar images. Decision support systems such as Google Analytics provide information and thus offer help with decision-making. Design decisions in data collection and derivation of new data will be done by the decision support system. Nevertheless, the interpretation of results is made by the users. In automated decision-making systems, the algorithm does all the work with the data and algorithm and presents the user with the final output. Netflix product recommendations is an example of an automated decision-making system [5].

Data is everywhere, and the uses are increasing and impacting society more and more. According to Bosch and Olsson [6], software, data, and AI are rapidly transforming conventional businesses. They have noted how the usage of data changes throughout the process of digital transformation. Further, they outline the steps companies take when moving from reactive use of data towards proactive use. As a first step, companies start using data for quality assurance and troubleshooting. Then in the next step, data is used for

internal improvement of product performance. Using the data collected from one customer, insights are drawn and delivered as value to the same customer in the later step. Data is considered as an asset in the subsequent step and comparative analysis is done and used for gaining better profits. In the final step, data from the original customer base are used to monetize with a second customer base. Often companies tend to appear in multiple steps at the same time. From this, it can be inferred that data dimension plays an important role in the evolution from a traditional to a digital company [6].

2.2 3Vs of Big Data

The characteristics of big data are marked by three Vs namely Volume, Velocity, and Variety. Volume refers to the amount of data, variety refers to the number of types of data and velocity refers to the speed of data processing [31].
Volume: Volume is an important component of the 3 Vs framework which is used to define the size of big data that is stored and managed by an organization. Volume refers to huge data in unimaginable sizes and unfamiliar numerical terms. Data is produced through human interaction with machines and networks on systems such as social media, sensors, and mobile devices.
Velocity: Velocity is a component of the 3 Vs framework that is used to define the speed of increase in big data volume and its relative accessibility. It is also known as data in motion. For instance, 200 million emails, 300000 tweets, or 100 hours of YouTube videos are created every minute. This increased velocity creates new challenges to consistency and completeness for big data collection.
Variety: Variety is a component of the 3 Vs framework which is used to define the different data types, categories as well as associated management of a big data repository. The massive amount of heterogeneous data is gathered from sensors, social media, and wireless networks in the form of interactive data such as website logs, astronomical data, medical records, etc. Understanding and extracting insights from such huge diversity demand high processing power and accurate data processing algorithms.

The challenges of big data management result from the expansion of all three characteristics. The amount of data that can be extracted from the

digital universe is continuing to expand as users come up with new ways to scrub and process data.

2.3 Synergy between Data and Artificial Intelligence

AI's ability to work well with data analytics is the main reason for data being an integral part of AI. AI algorithms like machine learning and deep learning are capable of mining every small detail from the input data and those inputs are used to generate new rules to fulfill its function [32]. Data and AI are merging into a synergistic relationship, where AI is useless without data and data is insurmountable without AI. Big Data will continue to grow larger as AI becomes a viable option for automating more activities, and AI will become a bigger field as more data is available for learning and analysis. Moreover, business decisions are based on big data that previously were based on guesswork or painstakingly constructed models of reality [33]. The sheer volume and variety of data consumed by modern analytical pipelines have greatly strengthened the connections between data integration and machine learning. [34]. Data management systems are increasingly using AI models like machine learning to automate parts of data life cycle tasks. Examples include data cataloging and inferring the schema of raw data [35]. Data analytics drives nearly every aspect of our modern society, including mobile services, retail manufacturing, financial services, life sciences, and physical sciences [33]. In most industries, established competitors and new entrants alike will leverage data-driven strategies to innovate, compete, and capture value from deep and up to real-time information [36]. However, in the current scenario organizations struggle with collecting, integrating, and managing the data. AI will not solve these data issues rather it will only make them more noticeable.

2.4 AI-enhanced Embedded Systems

Currently, considerable transitions are ongoing in the embedded systems industry, i.e. markets becoming more fast-changing and unpredictable, customer requirements becoming increasingly complex, rapidly advancing technologies,

and the constant need to shorten the time-to-market of new products [37]. Moreover, while the ability to manufacture high-quality mechanical subsystems remains perilous, it is no longer the key identifier and what makes a company competitive. During the last decade, along with electronics and software, AI has been introduced into many products, and embedded systems companies are becoming increasingly AI-driven [29]. AI/ML is becoming a horizontal technology: its application is expanding to more domains. Embedded Systems is also increasingly integrating AI into applications for performance improvement. Applications that involve both "traditional" software and Artificial Intelligence components are referred to as AI-enhanced Embedded Systems throughout this thesis. For instance, an embedded system that uses sensors to monitor things like temperature and vibration. Such a system should be able to detect anomalies in the early stages of things starting to go wrong, make predictions about future events, and alert its human supervisors as to what's going on. Here, AI is not the key component that controls the whole system, but it is used to enhance the performance of the entire system. Since AI-enhanced embedded systems rely heavily on software, it is expected that Software Engineering methods and tools can help. However, the development differs from the development of "traditional" software systems in a few substantial aspects. Hence, traditional SE methods and tools are not sufficient by themselves and need to be adapted and extended. AI-enhanced applications and AI-intensive applications are very common in the online domain. However, in the Embedded System domain mechanical subsystems, electronics and software are integral parts of embedded systems. Consequently, the developers won't be experts in AI application development which in turn makes integration of AI components difficult. Moreover, the data will be generated by both software as well as AI components. Thus, volume, velocity, and variety of data increase and should be managed accordingly to reap maximum benefits from the data.

2.5 Data management for AI-enhanced Embedded Systems

Data management is an administrative process that includes acquiring, validating, storing, protecting, and processing required data to ensure the accessibility, reliability, and timeliness of the data [38]. Inappropriate treatment of

data leads to data becoming corrupt, unusable, or completely useless. Companies trying to become data-driven are increasingly collecting and storing data from all possible sources. However, such companies need to understand that simply collecting data is not enough instead there is the need to understand from the start that data management and data analytics will be successful only after putting some thoughts on how to gain value from the collected raw data [32]. Efficient systems for processing, storing, and validating data, as well as effective analysis strategies are required beyond data collection. Each step of data collection and management must lead towards acquiring the right data and analyzing it in order to get the actionable intelligence that is required to make data-driven decisions [39]. Managing the data is the first step towards handling the large volume of data, both structured and unstructured, online and offline that floods daily. Data management best practices enable organizations to harness the full power of the data and gain the insights needed to make the data useful [40].

When designing artificial intelligence solutions, practitioners spend a significant amount of time focusing on aspects such as the nature of the problem, selection of learning algorithms, etc. However, little attention is often provided to the data on which the AI solution operates. As it turns out, the characteristics of the data are one of the absolute key elements that determine the right models for an AI solution. One possible reason for this indifference is that significant research has been done on data management practices over years. However, data required for AI models need to undergo substantial preprocessing before feeding it to the models. Moreover, the volume, variety, velocity, and veracity of data is increasing daily which acts as a compelling reason for the development of data management practices specifically for AI models.

2.6 Agile Methodology

Huge amounts of digital data are being generated through various sources such as sensors and devices. A significant part of it gets stored in hopes of finding ways to use it and generate useful insights from it. However, the exponential growth of data has rendered most of the attempts to analyze it inadequate. Besides, the unique approaches we use to acquire, preprocess, store, process big data, and generate the desired results, impose high demands on the ap-

plications performing these tasks [41]. The companies working with data are focusing on storing and processing large amounts of information [42]. The goal is not just to be able to process big data, but also to arrive at useful conclusions that are accurate and timely. A systematic approach similar to SDLC is needed for the development of data products considering the distinctive characteristics of big data and the available infrastructures, tools, and development models. Currently, data management practices struggle to keep up with the high velocity of data and growing demands of real-time analytics which leads to poor data quality, and consequently trust in the data is compromised. Agile practices and philosophy solve the issues inherent in the highly linear approach [43]. Details of the end-state analytics models become clear only when the results meet the needs of the organization. Due to the experimental nature of analytics development, detailed requirements cannot be set with complete confidence. By adopting the agile philosophy for analytics development, results are expected to be shared more frequently to form a feedback loop of stakeholder opinion and use those needs to validate the current state and influence its evolution to an agreeable end-state [44].

2.7 DevOps

DevOps is a set of practices intended to reduce the time between committing a change to a system and the change being placed into normal production while ensuring high quality [45]. As with all technological revolutions, DevOps practices impact processes, products, associated technologies, organizational structures, and business practices and opportunities. By leveraging DevOps methodologies, teams have achieved speed, quality, and flexibility by employing a delivery pipeline and feedback loop to create and maintain software products. DevOps enable developers to initiate builds any time during the day, and the results are quickly available [46]. Best practices along with the overall mindset from DevOps can bring these same improvements to data analytics.

2.8 DataOps

DataOps which began as a set of best practices has now matured to become a new and independent approach to data analytics [47]. Companies are increas-

ingly collecting data but are often failing to deliver insights on time. Dimensions of data such as volume, velocity, variety, and veracity are increasing day by day making data management a critical bottleneck. As a result, companies started to adopt a new data management approach called DataOps which is a set of practices that bring speed and agility to end-to-end data pipelines process, from collection to delivery [48]. Thus, DataOps is designed to solve challenges associated with inefficiencies in accessing, preparing, integrating, and making data available. DataOps is a method of managing data, with a greater focus on automation, communication, and integration. To manage collaboration and innovation, DataOps introduces Agile Development into data analytics so that data teams and users work together more efficiently and effectively. In Agile Development, the data team publishes new or updated analytics in short increments called "sprints." With innovation occurring in rapid intervals, the team can continuously reassess its priorities and more easily adapt to evolving requirements, based on continuous feedback from users. In DataOps, the flow of data through operations is an important area of focus. DataOps orchestrates, monitors, and manages the data in the company [48]. DataOps life cycle shares two active and intersecting data pipelines: Value pipeline and Innovation pipeline. DataOps automates orchestration and monitors the quality of data flowing through the Value Pipeline [49]. Innovation pipeline introduces new insights/value into the value data pipeline and is comparable to the DevOps framework. Value pipeline or data factory is a data pipeline that processes data and creates insights or value from it. From the definition of DataOps itself, it can be observed that data pipelines are the core elements that enable automation and orchestration of data [50]. Many companies struggle with the adoption of DataOps and difficulties in constructing data pipelines are one of the reasons. Therefore, the study on data pipelines is essential in order to reach the goal of better data management. Data pipelines are a popular concept in both academic and industrial communities. However, the development and maintenance of data pipelines is still a struggle for many companies. Data silos within the organization lead to teams having their own data pipelines and knowingly or unknowingly they do the same activities multiple times. Similarly, the data storage is affected as redundant intermediate data is stored by different teams. Although DataOps intend to solve the problem with communication, integration, and automation, it needs the help of data pipelines to realize its goals. On the other hand, due to a lack

of proper communication, data pipelines with the same activity with different names occur multiple times. To solve these problems, it is essential to have a domain-specific language for data pipelines that can be used within and across organizations.

2.9 Data Pipelines

The management of data is best captured using its data pipeline. A data pipeline is a set of tools and activities for moving data from one system with its method of data storage and processing to another system in which it can be stored and managed differently. Moreover, pipelines allow for automatically getting data from many disparate sources, then transforming and consolidating it in one high-performing data storage [51]. Data Pipelines are a chain of activities that are connected where each activity represents an atomic data task. Developing data pipelines enables the automation of most of the tasks in the data lifecycle. A data pipeline can be a simple process of data extraction and loading, or, it can be designed to handle data in a more advanced manner, such as training datasets for machine learning. Data pipelines are highly beneficial as they can process data in multiple formats from distributed data sources with minimal human intervention, accelerate data life cycle activities, and enhance productivity in data-driven enterprises [52]. Data pipelines enable traceability, fault-tolerance, and reduce human errors through maximizing automation thereby producing high-quality data [53]. However, a powerful argument against constructing a data pipeline is the cost of building and maintaining it, in terms of time, money, morale, and lost opportunities. Building a data pipeline demands specialized skills, time, and extensive experience in data engineering. Data pipeline construction is a task for which most data scientists have limited aptitude, interest, or training. Approximately 80% of an average data scientist's time is spent constructing data pipelines [54]. An alternate option is to buy a ready-made data pipeline from external vendors. As the use cases, organization culture, the expertise of the employees, etc varies from one company to another, it is always better to design a tailor-made data pipeline that can meet the requirements of the company. Automated data pipelines allow simple and flexible integrations, pipeline transparency, and automated workflows and processes to support even the most aggressive data management plans thereby delivering flexibility, scale, and cost-effectiveness.

2.10 Summary

This chapter presents eight main concepts discussed in this thesis, data products and applications, data and AI, AI-enhanced embedded systems, data management for AI-enhanced embedded systems, agile methodology for data, DevOps for data, DataOps, and Data Pipelines. Data products and applications give an overview of the significance of data and its applications at the industry level. This is relevant for all discussions presented in Chapters 4, 5, 6, 7, and 8. The data and AI section details the symbiotic relationship between AI and data. The third section on AI-enhanced embedded systems helps the reader to understand this new term and how it is different from AI-intensive, AI-powered, AI-enabled systems. Besides, it also details the unique challenges faced by practitioners working in the Embedded System domain while they develop and integrate the AI components in their software-intensive mechanical systems. Data management for AI-enhanced embedded systems explains the concept of data management, the practical challenges with data when the experts build data products. This section is important to understand chapters 5, 6, 7, and 8. Agile methodology for data describes the need for switching from the traditional waterfall approach to the agile approach while building data products including AI-enhanced embedded systems. DevOps for data is a section that describes what advantages data analytics can gain through adopting the best practices of DevOps. These two sections are closely related to the discussion on DataOps and thus required to understand chapters 5 and 6. Agile methodology and DevOps are used in DataOps and are two important techniques used in the current scenario. This discussion is relevant for Chapters 5 and 6. We provide a brief overview of data pipelines which is relevant to the discussion in Chapters 6, 7, and 8.

CHAPTER 3

Research Methodology and Design

Research methodology is a systematic way to solve a research problem. This study examines the need for an improved data management practice in the Embedded Systems domain and its effect on the quality of data products. Previous studies have demonstrated that practitioners are facing significant challenges in the development and maintenance of data products including AI models. The main objective of this study was to develop a data pipeline model with maximum automation that can be used for managing data to obtain high-quality data products. This chapter is divided into sections addressing the choice of research design, research questions, and motivation, selection of informants, data collection procedures, data analysis, an overview of the research process, and threats to validity.

3.1 Research Questions

The goal of this research is to empirically identify the data management challenges encountered during the development and maintenance of AI-enhanced embedded systems, propose an improved data management approach and empirically validate the proposed approach. We have adopted a qualitative re-

search methodology for addressing the research goals. The research is focused on the three primary research questions.
RQ1: What are the challenges associated with data management in embedded system companies?
RQ2: How can practices such as DataOps and Data Pipelines help address data management challenges?
RQ3: What implications does AI have on data management and what practices can help address the development and maintenance of AI-enhanced embedded systems?

The first research question (RQ1) aims to identify the data management challenges encountered by practitioners due to the recent AI advancements. Data management is a significantly explored topic and there exist many papers that discuss various data management practices. Therefore, a study to illustrate the challenges faced by practitioners with the new AI advancements is necessary to establish the need for a new/improved data management practice. The second research question (RQ2) was set to analyze how the practices like DataOps and Data pipelines help to address data management challenges in embedded system companies. This question explores the data management practices that evolved and analyses how the data management practices in embedded system companies adapted to the increasing significance of data. Moreover, it identifies the challenges at each phase of the evolution and the measures taken to address them. The third research question (RQ3) seeks to identify the impact of improved data management practice on delivering better quality data products. It also investigates how the improved data management practice helps companies to accomplish automation and thus accelerate the development and maintenance of AI-enhanced embedded systems.

3.2 Qualitative Research

Qualitative research was designed to collect, analyze and explain non-numerical data such as text, audio, and video to understand concepts, opinions, or experiences. The key to understanding qualitative research lies with the idea that meaning is socially constructed by individuals in interaction with their world [55]. Qualitative data use words for presenting results instead of numerical data and qualitative research is thematic in nature [56]. We adopted

this methodology as it allows us to help construct new ideas for how to improve or fine-tune a product or a practice. It also enables constructing a theoretical framework that emerges from data gathered during the research and enables the explanation of the results in a coherent manner. Qualitative data sources include observation and participant observation (fieldwork), interviews and questionnaires, documents and texts, and the researcher's impressions and reactions. Qualitative research enables the researcher to glean richer information, gain more in-depth insights into the real-time practices, and understand the underlying perceptions [57]. The main research techniques employed in this type of research are individual interviews, group interviews (focus groups), observations, and documents [58]. We chose individual interviews, group interviews, and observations as they can produce a wealth of detailed information about a small number of cases. This increased the depth of the understanding of the cases. In our study, we wanted to explore the use cases at companies, analyze their problems and propose an improved practice. Therefore, qualitative methodology is suitable for this study.

3.3 Case Companies

The entire research was a collaboration with Software Center [59] where there is a cooperation between academia and companies. Software Center has 16 companies and 5 universities as strategic partners. Multiple companies from Software Center participated in our research. The embedded system companies working on Artificial Intelligence, especially machine learning/deep learning were selected based on their domain, maturity in the adoption of AI. All the companies wish to remain anonymous as the studies often describe technology limitations, errors and pitfalls, and limitations in current development practices. Below, we provide a brief overview of the companies that are primary case companies in the research. Primary case companies allowed collaboration through interviews, in-company workshops, action research, and weekly meetings. The remaining companies in the Software Center also contributed through in-company as well as cross-company workshops. Therefore, they are listed as secondary case companies. More details about the participant companies are presented in Chapters 4, 5, 6, 7, and 8.

Primary case companies

Companies A, B, C, D, and E are marked as Primary case companies as they actively participated in this research by allowing collaboration through interviews, workshops, interactive sessions, weekly meetings, and action research.

Company A is a developer of an artificial intelligence platform designed to make the production of commercially viable AI applications swift, methodical and scalable. The company's platform enables their clients ranging from startups to large-scale enterprises to pursue the benefits of integrating AI into their systems.

Company B is a multinational company within the telecommunication industry that distributes easy-to-use, adaptable, and scalable services that enables connectivity.

Company C is from the automobile domain manufacturing their cars and does analytics based on the data from multiple manufacturing units, delivery units, and repair centers for identifying poor performing models.

Company D focuses on automotive engineering and depends on company C and does modular development, advanced virtual engineering, and software development for them.

Company E is within the manufacturing domain having more than 19,000 employees and they manufacture and market pumps. They have standards in terms of innovation, efficiency, reliability, and sustainability.

Secondary case companies

Companies F to N also contributed to the research through cross-company workshops. The reflections from the informants from these companies have helped in confirming the identified challenges and validity of the solution.

Company F works as a sales engagement platform that primarily enables and optimizes communication between sales representatives and potential prospects. Sales communication occurs in natural language via different communication channels, including emails.

Company G is a multinational technology company that develops, manufactures, licenses supports, and sells computer software, personal computers, consumer electronics, and services.

Company H is a global software company that develops both software and hardware solutions for home consumers.

The company I is a multinational automotive manufacturer and supplier of transport solutions. As the company's products are continuously growing in complexity and software size, the company is looking for strategies to prioritize their R&D effort and deliver more value to their customers.

Company J is a global car manufacturer that uses AI for building autonomous drive technology.

Company K is a global automotive manufacturer that collects and analyzes large amounts of data from the vehicle and hundreds of thousands of connected vehicles to develop increasingly more intelligent computer models that can identify patterns hidden from human view and capabilities.

Company L is a manufacturer of power tools, industrial and construction technology, and packaging technology. They apply big data and machine learning to their products and services to create AI solutions that are safe, robust, and explainable.

Company M is a multinational packaging industry that manufactures machines and materials for disposable packaging for milk, juice, and other liquid foods.

Company N is a manufacturer of network-based solutions in the areas of physical security and video surveillance. The company is active in many market segments, including transport, infrastructure, trade, banking, education, state and municipality, and industry.

3.4 Research Methods

Research methods are specific procedures for collecting and analyzing empirical data. Research methods are an integral part of the research design. The principal advantage of using qualitative research methods is that they force the researcher to delve into the complexity of the problem rather than abstract it away [58]. Empirical data is the information that is collected utilizing the senses, particularly by observation and documentation of patterns and behavior through experimentation to answer the research question [58]. Both data collection and data analysis can be qualitative as well as quantitative [60]. The qualitative data collection process entails the generation of massive amounts of data [61]. The audio- or video-recording data collection method is followed by the transcription before the data analysis [60].

Case Study

The case study is simply an in-depth study of a particular instance, or a small number of instances, of a phenomenon [62]. The goal of the case study is to study contemporary phenomena within their real-life context, especially when the boundaries between the phenomenon and context are not evident [63]. Advantages of the case study method include data collection and analysis within the context of phenomenon, integration of qualitative and quantitative data in data analysis, and the ability to capture complexities of real-life situations so that the phenomenon can be studied in greater levels of depth. Case studies do have certain disadvantages that may include lack of rigor, challenges associated with data analysis, and very little basis for generalizations of findings and conclusions [64]. Research strategies are classified as exploratory, descriptive, explanatory, and improving based on the purpose. A case study is a research method that was originally used primarily for exploratory purposes although it can be used for descriptive, improvement, and explanatory purposes [65]. For exploratory research questions, the case study strategy is a perfect match. However, also for descriptive research questions, the case study may be feasible if the representativeness of a sampling-based study may be sacrificed for better realism in a case study. If representativeness is critical, the survey is a better option. Explanatory research questions may be addressed in case studies, but the evidence is not a statistically significant quantitative analysis of a representative sample, rather a qualitative understanding of how phenomena function in their context. If quantitative evidence is critical, the experiment strategy is the better option. For improving the type of research purposes, the action research strategy is a natural choice, which we consider as a variant of case study research [65]. We have adopted an exploratory case study as we wanted to identify the challenges associated with the data management practices in the real-world company context. The exploratory case study also allowed us to capture the complexities of data management in the Embedded system company scenario.

Action Research

Action research gives the opportunity to work collaboratively with problem owners (concerned actors) at the organization and the possibility to propose, implement, and evaluate the solution in real-time. The Action Research ap-

proach typically means that researchers engage with a company over time and during a process. Problem owners are an inevitable part of action research since they share their skills, domain knowledge, and experiences [66] [67]. The main objective of action research in software engineering is to simultaneously solve a real-world problem and explore the experiences and results of problem-solving [68]. We chose the action research method for this study as the participatory aspect of it allowed us to systematically determine, define the problem with data management practices, and make a solution proposal in the context of an investigation. Moreover, it allowed us to actively participate in further steps of applying the solution in real-time which is termed as action [66] [67]. The action research process cycle consists of five stages namely (1) diagnosis, (2) action planning and designing, (3) action taking, (4) evaluation, and (5) specifying learning [66] [67]. Action research is advantageous as it has the potential to deliver robust and practical knowledge for a wide community of management and organization scholars [69].

We used a combination of case studies and action research for our study. Because initially, we wanted to identify the problems associated with data management practices in embedded system companies. We chose an exploratory case study as it allowed us to explore the complexities of the data management practices in the company context. Further, we wanted to implement and analyze the impact of the proposed data management practice. Therefore, we chose action research as it allowed us to focus on one aspect of the existing practice which is data pipelines that we wanted to improve. Moreover, through action research, we got the opportunity to implement the informed change which is an improved data pipeline model at the case companies, and observe the consequences.

3.5 Research Techniques

The research techniques are used to gather empirical data necessary to analyze the actions in real-world industrial settings [70]. Research techniques such as semi-structured interviews, literature reviews are appropriate for practical situations in which a fuller understanding of behavior, the meanings and contexts of events, and the influence of values on choices are useful for researchers.

Interviews

In interview-based data collection, the researcher asks a series of questions to a set of subjects about the areas of interest in the case study. Data collection through interviews is important in case studies [71]. The dialogue between the researcher and the subject(s) is guided by a set of interview questions. The interview questions are based on the topic of interest in the case study. That is, the interview questions are based on the formulated research question. The questions can be asked either to a group (focus group interviews) or individual practitioners. Questions that allow and invite a broad range of answers and issues from the interviewed subject are called open-ended while the closed offers a limited set of alternative answers. Interviews can be divided into unstructured, semi-structured and fully structured interviews [72]. In an unstructured interview, the interview questions are formulated as general concerns and interests from the researcher. In this case, the interview conversation will develop based on the interest of the subject and the researcher whereas in a fully structured interview all questions are planned and all questions are asked in the same order as in the plan. In many ways, a fully structured interview is similar to a questionnaire-based survey. In a semi-structured interview, questions are planned, but they are not necessarily asked in the same order as they are listed. We chose semi-structured interviews as they are helpful in the means of data collection because of two primary considerations. First, they are well suited for the exploration of the perceptions and opinions of respondents regarding data management issues and enable probing for more information and clarification of answers. Second, the opportunities for face-to-face contact with a researcher stimulate the interest in the project, establish a sense of rapport between respondents and the researchers [73].

Observation

Observation is the conscious noticing and detailed examination of participants' behavior in a naturalistic setting [74]. Observations can be conducted in order to investigate how a certain task is conducted by practitioners. There are many different approaches to observation. One approach is to monitor a group of practitioners with a video recorder and later on analyze the recording, for example through protocol analysis [75] [76]. Another alternative is to apply a "think aloud" protocol, where the researcher is repeatedly asking questions

like “What is your strategy?” and “What are you thinking?” to remind the subjects to think aloud. Observations in meetings are another type, where meeting attendants interact with each other and thus generate information about the studied object. An alternative approach is where a tool for sampling is used to obtain data and feedback from the participants [77]. While experiencing what is going on in a research site, researchers need to observe this and make detailed notes, called field notes, about the people, the concept they discuss, and the interactions that occur [74]. Participant observation was performed and field notes were taken during the action research. Observation as a data collection method can be structured or unstructured. In structured or systematic observation, data collection is conducted using specific variables and according to a pre-defined schedule. Unstructured observation, on the other hand, is conducted in an open and free manner in a sense that there would be no pre-determined variables or objectives [65]. The unstructured observation was used in this research as the observation mainly happened during the weekly stand-up meetings, pair-programming, and weekly presentation of results.

Multi-vocal literature review

The multi-vocal literature review is used to explore and summarize existing evidence concerning a particular topic [78] [79] [80] and to identify gaps and limitations of existing practices. A Multivocal Literature Review (MLR) is a form of a Systematic Literature Review (SLR) [81] which includes the grey literature (e.g., blog posts, videos, and white papers) in addition to the published (formal) literature (e.g., journal and conference papers) [79]. MLRs are useful for both researchers and practitioners since they provide summaries of both the state-of-the-art and –practice in a given area. Grey literature by the practitioners was ignored tagging them as "unscientific" while practitioner interviews are done and reported by researchers have, for long, been considered as academic evidence in empirical software engineering. MLR is developed to lift such a double standard by allowing rigorously conducted analysis of practitioners’ writings to enter the scientific literature [79].

We employed semi-structured interviews, observation, and multi-vocal literature reviews as research techniques. For this research, we wanted to collect empirical evidence about the challenges associated with existing data management practices from the practitioners. We chose semi-structured interviews

as it allows informants the freedom to express their views on their terms. Moreover, semi-structured interviews allow us to gather in-depth, comparable, and reliable empirical data. One of the data management practices we identified was relatively new and there was not much peer-reviewed literature that discussed it. Therefore, we chose a multi-vocal literature review to frame a definition for that particular data management practice. We used unstructured observation as a research technique as we were allowed to attend the weekly team meetings and other discussions. Thus, notes were taken during the weekly stand-up meetings, pair-programming, and weekly presentation of results.

3.6 Data Analysis

Qualitative research yields mainly unstructured text-based data. These textual data could be interview transcripts, observation notes, diary entries, or medical records. In some cases, qualitative data can also include a pictorial display, audio or video clips (e.g. audio and visual recordings of patients, radiology film, and surgery videos), or other multimedia materials. Therefore, the data analysis methods should be a dynamic, intuitive, and creative process of inductive reasoning, thinking, and theorizing.

Qualitative Data Analysis

Data analysis in qualitative research is defined as the process of systematically searching and arranging the interview transcripts, observation notes, or other non-textual materials that the researcher accumulates to increase the understanding of the phenomenon [82]. The process of analyzing qualitative data predominantly involves coding or categorizing the data. Coding merely involves subdividing the huge amount of raw information or data and subsequently assigning them into categories [83]. Thematic coding using the NVivo tool and open coding are the two types of coding used in this licentiate thesis. Thematic coding is a type of qualitative data analysis that finds themes in the text by analyzing the meaning of words and sentence structure. As NVivo is a thematic analysis software that helps you automate the data coding process, there was no need to set up themes or categories in advance [84]. Open coding is a manual coding technique that starts from scratch and creates codes based

on the qualitative data itself. Codes are manually created in such a way that it covers the entire transcript. These codes are then applied to the remaining transcripts and necessary adjustments are made so that the codes apply to all transcripts in the study [85].

3.7 Research Design

The first and foremost step in the research process was to discover an idea for research. The research idea originated from the conclusions of a study that was conducted together with one of the other team members for identifying the software engineering challenges for AI-enhanced systems in the embedded systems domain.

Our primary intention was to empirically identify the data management challenges encountered during the development and maintenance of AI-enhanced embedded systems, propose an improved data management approach and empirically validate the proposed approach. The RQs are formulated in such a way that they start with problem identification, progress through analyzing the existing approaches, identifying the challenges with the current approaches, proposing a solution, and validating the proposed solution. We selected qualitative research as it enables us to learn about the practitioners' perspective on the current situation, as well as the practitioners' willingness to make the transition to a new data management approach. Besides, it helps us to identify the needs of the practitioners, the problems they face with the existing data management approaches, and the impact of those problems on the final product. Further, qualitative analysis facilitates the generation of ideas for improvements in the data management approach. Quantitative research methodology can measure behaviors and helps to answer questions such as "how often" and "how many". However, our intention here was to explore the data management challenges, current practices, evolution of these practices, etc. Collecting data from informants is the best possible technique to understand the challenges in the embedded system industries. In quantitative methodology, free text responses can not be permitted and consequently, contextual detail might be missed. Therefore, we selected a qualitative research methodology for our research. The overview of the research process is as shown in figure 3.1

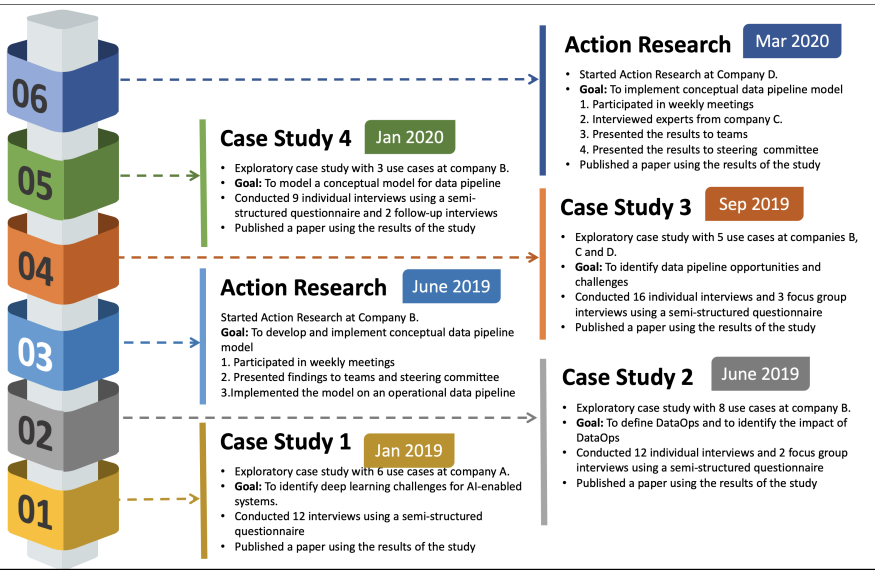

Figure 3.1: Research Process Overview

Case Study 1: Problem Identification

To learn more about the topic under investigation, a literature review was conducted which enables the researcher to identify similar studies that have been conducted in the past. Through literature review, the researcher identified papers on data management challenges. However, all the previous studies were focused on machine learning-based data products while deep learning is also an increasingly used technique in industries. Therefore, the researchers decided to conduct the study on data management challenges for deep learning models. To explore and identify the challenges encountered by practitioners during the development and maintenance of data products using deep learning techniques, a case study method was used as it allows for the examination of the phenomenon in depth using various kinds of evidence obtained from interviews with those involved, direct observation of events and analysis of documents and artifacts. Interviews were the main technique for data collection as illustrated in Table 3.1. We adopted an exploratory case study method and conducted 12 in-depth interviews with practitioners from 6 different do-

mains at company A. The selected informants also provided documents to better understand their use case description, data formats, Deep Learning model description, etc. All these interviews were semi-structured where the questions were open-ended. Because we intended to collect more in-depth information from the informants. Moreover, semi-structured interviews are a great way to delve deeper into issues. All the interviews except one were conducted face-to-face. One of the practitioners was working from home and so he appeared through video conference. With the permission of the practitioners, all the interviews were recorded and transcribed later. Moreover, empirical data was collected through observation during 2 cross-company and 5 in-company workshops. With the audio, video recordings, and notes made from interviews, observation, and document analysis, the categorization of challenges was done and sent to the practitioners for review. Data analysis was performed using the NVivo tool for thematic coding after transcribing the interview recordings. The codes were analyzed and results were formulated. The reflections of the participants were recorded and the researcher conducted follow-up interviews with 3 senior practitioners to clarify the doubts. The final results were sent to two senior practitioners and changes were made according to their comments. The results were then published as a paper and also presented in 9 workshops of which 5 were cross-company workshops and 4 were academic workshops.

Case Study 2: Analysing the existing Practices

The next step in the research was to investigate how the data management practices such as DataOps and data pipelines help address the data management challenges. We conducted an interview study with multiple teams working on 8 different use cases from case company B. Two focus group interviews and 12 in-depth semi-structured interviews were conducted for this study. The researcher was also allowed to attend the weekly stand-up calls and meetings. Further, the empirical data collected through observation during 2 cross-company and 3 in-company workshops. Therefore, observation and field notes also contributed significantly to this study. Case company B was trying to practice DataOps as it is an efficient method to improve the quality and reduce the cycle time of data analytics. As DataOps is a new concept, there was not much peer-reviewed literature on DataOps. Therefore, a multi-vocal literature review was conducted incorporating grey literature as well as peer-

reviewed papers. The results obtained through interviews, observations, field notes, and multi-vocal literature review were presented before the practitioners involved in the study and also to the steering committee of the company to get the approval. The reflections and comments from the informants were incorporated into the final results and published as a paper.

Case Study 3: Analysing the existing Practices

The previous study on DataOps showed that the data pipeline is the critical element. To explore the data pipeline management challenges, we used a case study method for which we used four data pipelines from 3 case companies. Because, we wanted to perform in-depth, multi-faceted explorations on data pipelines in a natural real-world context. We prepared an interview guide and conducted 16 semi-structured interviews with practitioners from 3 companies who were working on 5 different data pipelines. One of the researchers was an action researcher at two of the companies and was allowed to attend the weekly stand-up meetings. Therefore, the researcher prepared field notes during those meetings which are also used along with the interview transcripts. The empirical data was also collected through observation during 4 in-company and 2 cross-company workshops. All the interviews, meetings, and workshops were video-conferencing due to the COVID-19 pandemic. The interview transcripts and field notes were coded using the open coding technique and the results were formulated. The results were presented before the steering committee at two companies for getting the approval. Further, the results were presented before the other teams in the organization to get some external reflections. These reflections are also incorporated in the final results and a paper was published with this data. The results from the paper are presented in 3 cross-company workshops and 5 in-company workshops.

Case Study 4: Propose a Solution

Further, we wanted to develop an improved data management approach that can support the development of high-quality data products especially AI-enabled systems. To serve this purpose, based on the insights from the previous study, we developed a conceptual model for data pipelines. Interviews were the primary technique for data collection. We conducted 9 in-depth interviews as well as 2 focus group interviews to understand more about the

existing data pipelines at one of the case companies. Moreover, the researcher collected data through observation during various meetings inside and outside the company and prepared notes. With these notes together with the meeting insights and interview transcripts, a conceptual model for data pipelines was developed. To validate the conceptual model, the in-company presentation was done to collect comments from teams involved in the study. Then another round of presentation was done to collect reflections from teams working on data products but were not involved in the study. Further, the model was presented at two other companies for external validation and incorporated their comments. The model was published as a paper and presented in 2 cross-company workshops and 4 in-company workshops. Overall empirical efforts on exploratory case study research are presented in table 3.1

Action Research: Implement and Validate the Solution

Furthermore, the study sought to maximize automation and minimize human intervention through the implementation of the conceptual model for the data pipeline. Action Research at company B and D contributed to the results of this study. The realization of the conceptual model was done through action research in the case of company B from the telecommunication domain. Although we planned to implement the model at company D, due to company restrictions, it was postponed. The literature review was performed to understand the working of fully automated Data Pipelines. Because literature review is an excellent methodology through which we can identify the data pipelines that are implemented in various industrial domains. IEEE Xplore, ACM Digital Library, Web of Science, and Google Scholar were the main source for Literature review.

At company D, the researcher together with a data analyst and superuser developed a modification plan for one of the existing data pipelines and presented it before the steering committee. The modification plan was then presented to case company C as company D is dependent on C. Unfortunately, the plan was rejected due to company restrictions. At company B, the researcher together with the help of a Data Scientist and Software Developer identified the data pipeline for modification. Further, a modification plan was submitted to the higher authorities and presented before the steering committee for approval. Document analysis was performed to understand the underlying architecture of the data pipeline. Due to time constraints and

other company restrictions, the researchers chose a small slice of the data pipeline and modified it according to the conceptual model. The implementation was accomplished through pair programming in which the researcher and software developer worked together. Pair programming improves design quality, reduces defects, reduces staffing risk, enhances technical skills. The results of this study are presented in two cross-company workshops and three in-company workshops. Overall empirical efforts are presented in table 3.2 that summarizes the stages of the action research cycle, total efforts, and the total number of interactions involved throughout the research. Figure 3.2 illustrates the research questions, research methods adopted, primary participant companies involved, and the results of the study.

Table 3.1: Overview of Exploratory Case Studies

Research Question	Research Objective	Research Technique	Data Collection Technique	Case Companies and Use cases	Duration	Interviewed Experts - Roles
What are the challenges associated with data management in embedded system companies?	To identify the data management challenges	1. Interviews 2. Observation	1. Semi-structured Interviews (12) 2. Observation and field notes from weekly meetings 3. Observations from 2 cross company workshops and 5 in-company workshops	Case companies (6) Use cases (6)	Jan 2019-March 2019	Principal Data Scientist (2), AI Research Engineer (2), Data Scientist (3), Head of Data Analytics team
How can practices such as DataOps and Data Pipelines help address data management challenges?	To identify the existing data management approaches	1. Interviews 2. Observation 3. Multi-vocal literature review	1. Semi-structured Interviews (12) 2. Focus group interviews (2). 3. Observation and field notes from weekly meetings 4. Observations from 2 cross company workshops and 3 in-company workshops	Case companies (1) Use cases (8)	June 2019-December 2019	Senior Data Scientist, Integration and Operations Professional, Analytics System Architect, Data Scientist, Senior Customer Support Engineer, Developer, Senior Data Engineer, Program Manager
	To identify data pipeline management challenges and opportunities	1. Interviews 2. Observation	1. Semi-structured Interviews (16) 2. Focus group interviews (3) 3. Follow-up Interviews (2) 4. Observation and field notes from weekly, biweekly meetings 5. Observations from 4 in-company workshops and 2 cross-company workshops	Case companies (3) Use cases (5)	September 2019-February 2020	Senior Data Scientist(3), Data Scientist(2), Analytics System Architect, Software Developer(3), Senior Data Engineer, Data Engineer(2), Data Analyst and Superuser, Director of data analytics team, ETL developer, Product Owner for data analytics team
	To develop a conceptual model for data pipelines	1. Interviews 2. Observation	1. Semi-structured Interviews (9) 2. Follow-up interviews (2) 3. Observation and field notes from weekly meetings 4. Observations from 4 in-company and 4 cross-company workshops	Case companies (1) Use cases (3)	January 2020-May 2020	Senior Data Scientist(3), Data Scientist(3), Analytics System Architect, Software Developer(2)

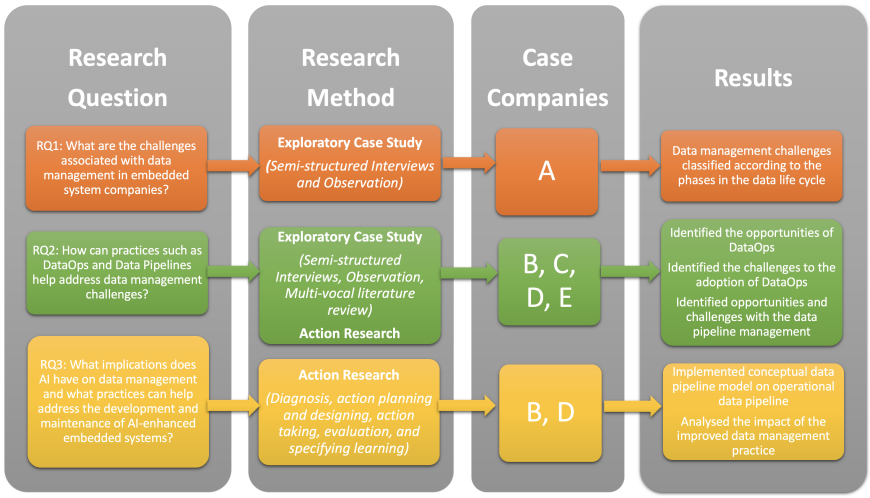

Figure 3.2: Overview of Research Questions, Research Methods, Companies and Results

3.8 Threats to Validity

This section discusses threats to validity regarding how our research questions were answered.

Construct Validity

Construct Validity includes two components: the measure should be exhaustive, and the measure should be selective in that it only covers aspects of the target theoretical construct. To ensure construct validity, a few cases were excluded from the results as some of the interviewers did not have a proper understanding of the discussed concepts. As a result of the screening process, our study has some limitations with several interviews. However, this limitation can be counted as an opportunity for further inquiry in future works. For reducing researcher bias, the interviews were conducted by a minimum of two researchers. Further, before the interviews, we developed the semi-structured interview guide and distributed it among the interviewees. A short description of the topic to explore is sent to the interviewees before the interview.

During the interview, we again explained the topic of the study as an introduction. We rephrased the question whenever the response becomes off-topic or asked them to elaborate when we received ambiguous answers. Further, while analyzing the transcripts if there are any confusions or lack of clarity, we contacted the interviewees to resolve this problem.

Internal Validity

Internal validity is defined as the degree to which the observed outcome represents the truth in the population we are studying and, thus, is not due to methodological errors [66]. The results of this thesis could potentially be affected by this threat since the results and strategies associated with RQ2 and RQ3 were developed in the company context. As the researcher only had limited access to the descriptions of the strategies, it is not possible to investigate if other factors were more influential to the final result than the proposed strategies. To minimize internal validity threats, one of the co-authors, who has in-depth knowledge about the data processed in the company, was asked to validate the findings. Further, the findings were validated through the steering committee at the respective companies.

External Validity

The presented work is derived from the cases studied with different teams in the domains of manufacturing, automobile, and telecommunication. Some parts of the work can be seen in parts of the company differently. All the terminologies used in the companies are normalized and the implementation details are explained with the necessary level of abstraction [86]. We do not claim that the opportunities and challenges will be the same for industries from different disciplines.

3.9 Summary

This chapter discussed the five research questions addressed in this licentiate thesis. The research methods are discussed in general and how they were used to address each research question is addressed in the research design section. Overall empirical efforts are presented in two separate tables. We discuss the three main threats to validity and how we tried to mitigate them.

Table 3.2: Overview of Action Research Cycle

Action Research Cycle	When	Action	Fact Finding Data Collection	Reflection and Learning
1. Diagnosis	March 2020	Presenting the data pipeline challenges during in-company workshops	Collecting reflections from various teams in the companies.	Participants confirmed that the challenges are real
2. Action planning and designing	April 2020	1. Get steering committee approval	Presented the challenges before steering committee for approval	Most of the committee members supported the need for a solution that can better address the challenges
	April 2020	2. Collect data on existing fault-tolerant data pipelines implemented at large-scale industries	Literature review on fault-tolerant data pipelines implemented at large-scale industries like Google, Microsoft, Facebook and LinkedIn.	Analyzed large-scale industries like Google, Microsoft, Facebook and LinkedIn.
	April 2020	3. Scheduled meetings with teams to explore and understand the existing data pipelines in both the companies A and B	Participant observation and field notes Interviews at A - 16 Interviews at B - 10	Studied 3 data pipelines at Company A Studied 2 data pipelines at Company B
	April 2020 - December 2020	4. Organized workshops with company B and their data suppliers Organized workshop with company A	Workshop at B - 3 Workshop at A - 2	Data Pipelines at Company A directly delivers data product to their customers and consequently, data leaks through data pipelines forced them to appoint a data flow guardian
	April 2020 - December 2020	5. Attend weekly meetings at companies A and B	Meetings at A - Weekly once Meetings at B - Weekly once	Data drift is not sporadic Data flow guardian overloaded with alarms
	April 2020 - December 2020	6. Biweekly presentations to update progress and collecting reflections	Follow-up meetings at Company A - 7 Follow-up meetings at Company B - 5	Communication between Data suppliers and Company B was not healthy Multiple teams at company A were redundantly doing same data pipeline activities
3. Action taking	May 2020- June 2020	Presented the conceptual data pipeline model	Collected reflections/comments from the participants at A and B	The model was acceptable for the teams at both A and B with minor modifications according to their use cases
	May 2020- June 2020	Approval for implementing fault detection and mitigation components	Steering committee approved at A Steering committee postponed the implementation at B	Data pipeline itself is not matured at Company B compared to A. Dependency of Company B to external company restricted them from implementing fault detection and mitigation components
4. Evaluation	June 2020	Find the data pipeline link where the fault detection and mitigation strategy should be coded without hampering the existing data flow through the pipeline as the data pipeline is operational	Familiarized HBase coding with Python Number of missing data dumps were noted prior to the implementation. Checked the effect of fault detection and mitigation	Failed dump IDs during last 30 days were 32453 which constituted 37% of total data dumps.
5. Specific Learning	July 2020	Conduct follow-up meeting to give a final presentation of results	Follow-up meetings at Company A - 2 Follow-up meetings at Company B - 3	Identification of faults and corresponding mitigation strategies at each step of the data pipeline Incorporation of automatic fault detection components and mitigation strategies in a small slice of a data pipeline. Benefits and limitations of fault detection component and mitigation strategies in the data pipeline.

CHAPTER 4

Data Management Challenges for Deep Learning

This chapter has earlier been published as
Data Management Challenges for Deep Learning
Munappy, A. R., Bosch, J., Olsson, H. H., Arpteg, A., Brinne, B
2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 140-147). IEEE.

Over recent years, deep learning has reached the pinnacle of popularity due to its ability to learn deep representations. The capability to learn multiple levels of representations and abstractions from data makes it unique among machine learning techniques[87]. It has been used successfully in image classification[88], object detection[89], natural language processing and information retrieval[90]. Even though the terminologies like machine learning and deep learning are used interchangeably, they do not refer to the same concepts. Machine learning requires a significant amount of work spent on feature engineering[91]. However, deep learning is a particular type of machine learning technique more refined Artificial Intelligence technique that can learn from unlabelled data which is an attractive feature demanded by most of the real-world applications[92]. Even though deep learning models have remarkable

abstraction and generalisation capabilities, these systems are data hungry in nature. i.e a massive amount of data is required to train Deep Neural Networks. As the requirement for a large amount of data is significant, large-scale data management issues arise in collecting, processing, analysing, sharing and deploying datasets. Although deep learning models are extensively used in a variety of applications, data management for deep learning has received limited attention from researchers and practitioners.

Over the years, there has been a significant advancement in deep neural networks and algorithms. However, this advancement has not been matched with similar progress in data management. Therefore, there is a strong need for new techniques and automated tools to be designed that can assist practitioners in preparing and ensuring quality data throughout the data pipeline workflow.

In this paper, we discuss six real-world industrial applications of deep learning in different domains such as medical imaging, gaming, real-estate, manufacturing systems highlighting the key data challenges to data that can significantly impact the overall performance of DL systems. We do not aim to provide a comprehensive background on technical details and general application of deep learning (see e.g. [93], [94]) nor do we explain extensive challenges faced by real-world software-intensive systems as [26]. Instead, we focus on different data management challenges faced by DL experts while building DL application models. In this paper, we introduce a number of example applications of deep learning frameworks, we explain significant challenges and we categorise these according to the development phase in which it is encountered. The contribution of this paper is twofold. First, it presents the main data management challenges, that need to be addressed for developing high performance and operational deep learning models. Second, the paper classifies the challenges according to the phase in which they are encountered and identifies the main areas that requires attention.

The rest of this paper is organised into six sections. Section II is a description of the background and related works. In section III, we introduce the research methodology adopted for conducting the study. Section IV details the cases explored in the study. Section V focuses on findings of the case study and maps data management challenges encountered at each stage of the data pipeline with the use cases. Finally, Section VI summarises our conclusions and completes this paper.

4.1 Background

Deep learning[92] provides major advancements in solving the problems which were previously unbeatable by artificial intelligence and machine learning techniques. Due to this reason, it is being used in hard scientific problems like reconstruction of brain circuits [95], mutation analysis in DNA [96], structure-activity prediction of potential drug molecules[97] and online particle detection[98]. Deep neural networks are also opted to decipher many challenging tasks in speech recognition[99] and natural language processing[91].

Deep learning became the focal point after Krizhevsky et al. [100] demonstrated the remarkable performance of a Convolutional Neural Network (CNN) [100] based model on a challenging large-scale visual recognition task [101] in 2012. A substantial credit for the current reputation of deep learning can also be attributed to this influential work. Great contributions to deep learning research have been made by the computer vision community by providing solutions for the problems encountered in medical science to mobile applications since 2012. The recent breakthrough in artificial intelligence in the form of tabula-rasa learning of AlphaGo Zero [102] also owes a fair share to deep Residual Networks (ResNets) [103] that were originally proposed for the task of image recognition.

Data - The fuel

Data is the fuel for deep learning models. Massive datasets are used to train deep neural networks in order to mimic human intelligence. It is data which allows industries to stay on top of trends, provide answers to problems, and analyse new insights to great effect. There are numerous algorithms in deep learning tailored for various applications which deliver high performance. However, no algorithm can guarantee the same performance over all the datasets. This is a clear indication of the importance and effect of data in the performance of DL models.

Data Management

Data management for deep learning can be defined as a process which includes collecting, processing, analysing, validating, storing protecting, and monitoring data to ensure the consistency, accuracy, and reliability of the data. Deep

learning has been successfully applied in industry products that take advantage of the large volume of digital data. However, real-world data needs to be processed and managed before feeding it as input to the deep learning models. Training a deep learning model with such massive and variegated data sets is challenging and several aspects need to be considered. E.g. data sparsity, redundancy, and missing values. In order to ensure high performance of DL models, not only good algorithms but also the management of data is required. A set of good data management practices should be followed from data collection, through data processing and analysis, dataset preparation and deployment of the model.

As DL applications are highly data-driven, it could be benefited from data management and database techniques to accelerate the speed of training. Wang et. al[40] describes how certain challenges like data dependency, memory management, concurrency, data inconsistency can be solved by combining database techniques and deep neural networks.

DL models demand large volume and variety of data which relates it to the field of Big Data. Popular companies like Apple[104], Google[105], Facebook, Microsoft are collecting a copious amount of data on a daily basis through applications like Siri, Google translator, Bing voice search[106] to provide a variety of other services such as reminders, weather reports, personalised recommendations. Although big data offers numerous opportunities, it also imposes consequential engineering challenges[107]. X. W. Chen et al. describes the big data challenges such as streaming data, high-dimensional data, scalability of models, and distributed computing[65]. However, in these papers, deep learning is considered as a solution for management of data. Data management challenges involved in implementing deep learning models are not seriously considered and our paper intends to focus on that perspective.

4.2 Research Method

In order to set the scope for the type of empirical studies we address in this paper, an interpretive multiple-case study approach was adopted adhering to the guidelines by [108]. Usage of multiple cases should be considered same as the duplication of a study or an experiment which means that the inferences from one case should be compared and contrasted with the results from the other case(s). The objective of this study is to identify challenges specifically

related to the management of data in various real-world DL applications. The challenges identified are based on our interpretations of the experiences of experts who implement DL systems in a real-time scenario with real-world datasets. This type of case study research is appropriate as it facilitates the exploration of the real-life challenges in its context through a variety of lenses[108]. The overall research design and the major steps in the research process of the study are described below.

Expert Interviews

The objective of the study is to explore data management challenges encountered while implementing DL models in real-world settings. Each case in the study refers to a software-intensive system that incorporates DL components developed by an organisation. For the study, a sample pool of DL experts who works in seven different domains were selected by their expertise in the area of study. The selected seven practitioners include two authors of this paper. From the acknowledgment in the literature (and our experiences when soliciting interviewees), it can be inferred that only a few experienced practitioners are skilled in the area of intersection between DL and SE, Table 1 illustrates the vast experience of our interviewees in incorporating DL components across multiple domains.

Data Collection

Semi-structured interviews were used to acquire qualitative data. Based on the objective of research to explore data management challenges for deep learning systems, an interview guide with 40 questions categorised into four sections was formulated. The first and second sections focused on the background of the interviewee. The third section concentrated on the importance of data in various projects and the last section inquired in detail about data management, the challenges faced during every phase of the data processing pipeline. The interview guide was reviewed by the authors and some additional questions were added, a few similar questions were merged together and some totally irrelevant questions were removed finally forming an interview protocol with 20 questions spread across four different categories. All interviews were face-to-face except for one which was done via video conference and each interview lasted 45 to 55 minutes. All the interviews were recorded

with the permission of respondents and were transcribed later for analysis.

Data analysis

After the interviews, audio recordings of interview were sent for transcription and a summary of each interview was made by the first author highlighting the main focus points of the interview. The analysed points from the summary were cross-checked several times with the audio recordings and interview transcripts obtained after transcription. A theoretical thematic data analysis approach was opted for coding[109]. First, the author coded each segment of the interview transcript that was relevant to or captured something interesting about data in NVivo. In the first iteration, the aim was to identify the phases of data pipeline. After identifying the phases, a second iteration was performed to code the data management challenges encountered in each phase by setting high level themes as *(i) Data Collection, (ii) Data Exploration, (iii) Data Preprocessing, (iv) Dataset Preparation, (v) Data Testing, (vi) Deployment, (vii) Post-deployment*. The results deduced from the analysis were tabulated and sent to the authors for comments and then the final summary of the cases and mapping were sent to the interviewees for validating the inferred results.

4.3 Cases

This section describes different real-world DL cases that has been chosen for this research. All the cases reported here are using real-world dataset. A mapping between different data management challenges and projects is presented in a later section.

Recommender System

Many e-commerce and retail companies are leveraging the power of data and boosting their sales by implementing recommender systems on their websites. When a customer visits the website, the recommender system predicts users' interest and recommends electric products for them based on previous customer reviews and purchase history. Many times customers tend to look at the website for their recommendations. Personalised recommendations from the system would increase customer satisfaction and thus customer retention.

Table 4.1: Description of Use cases and Roles of the interviewees

Case	Use case of DL components	Interviewed Experts	
		ID	Role
A	Recommending products to the users in a personalised fashion	P1	Principal Data Scientist
B	Predicting the wind power using the historical weather data	P2	Data Scientist
		P3	Head of Data Analytics team
		P4	Data Scientist
		P5	AI Research Engineer
C	Estimating and predicting the price of houses	P2	Data Scientist
		P3	Head of Data Analytics team
D	Automated classification of skin lesions into benign and malignant	P2	Data Scientist
		P3	Head of Data Analytics team
		P4	Data Scientist
		P5	AI Research Engineer
D	Detecting the credit card frauds during gaming	P2	Data Scientist
		P3	Head of Data Analytics team
E	Predicting quality of paper boards	P4	Data Scientist
		P5	AI Research Engineer

In recommender systems, DL components are trained on user reviews and their purchase history.

"It's very difficult to focus on the things that aren't visible feature wise, such as tracking data. So that means that often you first develop the features the way you want them, and if you have time in the end you put tracking in, so you put the gathering of the data, that part of the code in. So that obviously means that it's not as well tested, and it's not as well tracked. It doesn't get the same love when you develop and so on. So that usually makes data quality bad."

Wind Power Prediction

Wind power is dependent on weather and so it is irregular and fluctuates over different time scales. Thus accurate forecasting of wind power can be considered as a major contribution for reliable large-scale wind power integration.

DL model is utilised to predict accurately how much electricity, how much power are all of the wind turbines going to generate within 24 to 48 hours so that an accurate report can be submitted to the power companies for which energy is supplied. The power companies have quite strict requirements like they have to accurately say how much power they are going to deliver and if not there are penalties that need to be paid if they don't manage to deliver the reported energy. A combination of wind and weather are predicted from which the power generated by the wind turbines can be calculated. The wind power is predicted based on the meteorological data obtained from the National meteorological agency. Deep Learning is used to forecast weather and thereby predicting the wind power that can be generated in the future.

"We have gotten our data in all sorts of different ways. When we got the data for the weather prediction case we actually got them on physical tapes. Those ... Really in boxes, with physical tapes. And then we had to digitalize them ourselves. So, I do not think there is any standard or framework to accomplish this. That is why data management itself is a problem when dealing with deep learning."

House Price Prediction

Predicting property values are of great interest to various parties in an economy. Estimation of house price is important to prospective homeowners, developers, investors, appraisers, tax assessors and other real-estate market participants, such as mortgage lenders and insurers. Real-estate investors and portfolio managers devise and carry out their investment decisions based on periodic evaluations of their real-estate portfolios. Individuals are interested in knowing the values of their properties before setting up their list prices. Tax authorities rely on the estimates of the properties' value as the basis for levying property taxes. Banks and mortgage providers conduct housing collateral valuation to qualify the borrowers for their mortgage applications. Initially, house price was predicted on the basis of comparison between cost and sale price and there were no accepted standards or certification process. So, the house price prediction model helps to fill up the information gap that existed before and also enhance the efficiency of the real-estate market. The house price prediction case was initially built on traditional assorted database system where SQL queries and data pipeline scripts were used whereas now it utilises deep learning technique where the model is trained with historical

sales data about properties, geography, and demography in the Swedish market. The house price prediction system is a long-running DL model deployed in production and is used by many banks in Sweden.

"Someone has to make a design choice, like is this interesting to collect, on what level, and what kind of metadata do I attach to it for example. Because it's easy to just log something, but then when we come later as data scientists and look at the data and we're like "okay, that's really good, but which user was that? Ah, we didn't log it". Okay, if you didn't log that, then I can't really combine that data with my other data set where I have it on user level. So I can't really, you know, all of those missing pieces are challenges"

Melanoma Detection

Melanoma is a type of skin cancer, which is not that common like basal cell and squamous carcinoma, but it has dangerous implications since it has the tendency to migrate to other parts of the body. Therefore, early detection can prevent it from spreading to other parts; otherwise, it becomes incurable. Deep learning bypasses all the complex methods of pre-processing, segmentation and low-level feature extraction. Although a lot of datasets like MEDNODE, ISIC Archive and many more are publicly available, dealing with real-world data is still challenging. Automated classification of skin lesions using images is a difficult task because of the unavailability of fine-grained varieties in the appearance of skin lesions. The skin cancer detector not only intends to detect whether a person has skin cancer or not but also what kind of cancer it is and thus how serious it is. Here, the deep learning model is used for diagnostic classification of dermoscopic images of lesions of melanocytic origin. Datasets are formed over several years by working in close collaboration with clinics. The company has restrictions on the use of the dataset and the requirement is even the data cannot leave the servers. With these restrictions, the practitioners adapt to the rules specific to the dataset and move the code and model to the server where data is stored for developing the DL model. The skin cancer detector is still not production ready.

"It is very difficult to scale the data collection, because you have to get something from a patient. Very intrusive things, like sticking electrodes into their skin and taking images, or something like that. So there, it is kind of hard to increase the amount of data you have quickly, because you need to see patients that go through the health care system, and you know. What you can

do is try to use publicly available data that is similar, but then you always had issues with the data not being quite the same. Not the same distributions, different cameras, different machines etc. So that is a difficult domain."

Financial Fraud Detection

Frauds in finance still amount for significant amounts of money. Around the globe, hackers and crooks are trying to find new ways of committing financial frauds. Therefore, trusting financial fraud detection systems programmed based on conventional rule-based method alone will not serve the purpose. This is where Deep Learning shines as a unique solution. The DL model uses customer details like payment history and activity history and payment request data such as payment method, amount location, etc. The post-payment signs of abnormal pay are also taken into account for detecting fraud. When it comes to modelling fraud detection as a classification problem, the main challenge comes from the fact that in real-world, the majority of the transactions are not fraudulent. However, in order to train DL models, counterexamples are also required.

"If you have a company that deals with credit card fraud or something like that, and then they record all the examples of when people have had the fraud. And then if they don't have the counter examples of the normal examples, then it's again difficult. "

Manufacturing Systems

Paper mill industry creates paper from pulp and then dry that into carton and cardboard which is further used for making milk cartons. A DL component is incorporated in the system to predict the quality of the resulting product based on all the measurements in the machine and measurements on the pulp that goes in. And there's also images of what's happening at the beginning of the machine, and images, microscope images of the fibers in the pulp. The company manufactures large quantities of paper board each year and wanted to minimise the material cost as much as possible while maintaining high quality. Quality of the paper board is predicted based on data from process sensors and images of wood fibers taken with the PulpEye technology. The DL models serve as a stepping point for controlling the manufacturing process so that the same quality could be maintained with less input material and waste.

"I think this data engineering in the beginning or, that is supposed to be in the beginning, has always turned out to be a much bigger problem than you think. Because you usually realize after you have started modeling that you have had some assumption that was not really correct, and then you have to go back and kind of redo the data engineering again."

4.4 Findings

This section presents a list of concisely described data management challenges encountered by practitioners while implementing DL components in real-world applications. Based on the study, we have identified seven stages through which data flows and the data management challenges raised in each phase. Our study is carried out with six use cases as mentioned above. Many of the challenges identified are use case specific and so a mapping between these use cases and data management challenges are shown in table 2.

Data Collection

The systematic process of gathering data from a range of sources relevant to the context is termed as data collection. Deep learning model should be constantly fed with data to continue improving performance while deployed in production-ready systems. Acquiring data is thus a crucial phase which needs attention.

Lack of metadata

Metadata is required for the practitioners as they might not be experts in the domain where they implement DL components. Practitioners mentioned that in many projects, lack of metadata creates confusion and poor understanding of the data. Due to poor organisation, the semantics of data is often obscured which in turn leads to ambiguities. When a dataset is handed in for building a stock market price prediction, the dataset may have different prices like opening price, closing price, quoted price, session price and without providing associated metadata information, it is hard for the practitioners to identify and distinguish different prices. Without metadata, it is not always possible to figure out if some pattern makes sense or not. If you know that a particular signal represents a temperature reading and it is always zero, then you know

it is wrong. On the other hand, if it is an on/off switch, then maybe it is zero all the time which is fine.

Data Granularity

Data aggregation may remove important data points which cannot be collected again. Even after collecting a huge amount of data and then aggregating it after a certain span of time will spoil the detail in the data. Thus fine granularity in data is lost through data aggregation techniques. Like in mobile networks, counter data is collected and aggregated some value over 15 minutes, and that's what gets saved. Because saving every second's data point is not affordable. And then in that aggregation, a lot of information is lost, which could mean that even though a lot of data are in place when looking at it in detail, granularity actually needed for a use case will not be there. So even if data is collected over ten years, the problem is still kind of limited by data collection choices which are difficult to get around. In our study, the recommender system case experience this data granularity problem. When the reviews from users are all logged for a long period and handed over for building recommender system, but failed to log the user's identity, the data granularity is lost. And it is not possible to combine that data with other data set on user level.

Shortage of diverse samples

Upon training, the deep neural network should be given all possible instances and varieties of data so that it will not fail on inputting unseen data in production. However, during data collection, many companies collect a large number of normal samples and fail to collect the counterexamples of data. The DL model needs to be trained with counterexamples as well. From our case study, one extreme example we got is that financial fraud detection cannot be developed only with the samples of fraud cases, it also requires normal transaction instances in the dataset. Deep learning models cannot learn the normal cases by themselves when only the abnormal samples are fed during the training which leads to weird outputs after deployment.

Need for sharing and tracking techniques

Sharing the collected data with the practitioners is required while implementing the Deep Neural Networks. There is no defined channel or medium for sharing the collected data. According to the size of the data, different people choose different means of sharing. Some companies may opt to share the data in the form of excel files over email, FTP server or even in the form of physical tapes. Two of the experienced practitioners identifies tracking as an important measure by which data quality can be assured. However, due to the tight limit on time and resources, often data tracking is not focused much or is kept at a least priority leading to poor data quality.

Data Storage

Deep learning systems are powerful in memorising each and every piece of information given to it. So the amount of training data has the biggest impact on the performance of the model. General Data Protection Regulation(GDPR) is a regulation in EU law to protect online personal data. GDPR is a set of legislative rules which impose restrictions on processing and storage of information. Major companies who focused on collecting and maintaining datasets are able to build better DL models to a certain extent. The problem with small scale companies is that they do not have clear knowledge on how to collect and store data complying to the rules of GDPR and there is no framework or protocol to help them to do data collection efficiently. In such cases, a certain percentage of revenue needs to be paid as a penalty for not following the regulations of GDPR which end up in the deletion of a huge portion of data they collected over time. Even though this is a problem experienced by only one case in the entire study, it is still important as it has significant legal and financial complications involved.

Data Exploration

Data exploration is analysing the distribution of different datasets and data fields, checking the number of outliers and existence of missing data, examining how to connect the data together and build up basically a dataset that can be fed directly into the model.

Statistical Understanding

When confronted with data that needs to be analysed, the first step is to carefully identify the distribution of data. Statistical understanding is much required for determining the distribution of data. Even with sufficient knowledge in statistics, it is challenging to identify the distribution of data. The normal distribution or Gaussian distribution is that nice, familiar bell-shaped curve. But, data comes from a range of devices out in the wild and there is no point in assuming an easy to handle normal distribution. For instance, consider an image processing application, to model the pixel values efficiently, assumption of Gaussian distribution is inaccurate as it violates the boundary properties. In such cases, models like BMM(Beta Mixture Model) is opted. Without clear knowledge of statistical distributions, it will become difficult to model the distribution.

Deduplication Complexity

Dataset often has a lot of duplicates, some with slight variations and some exact copies. So analysing the dataset for duplicates and deduplication is a complex task. For example, consider a song recommender system trained on a dataset of songs. If you take a random song, there can be 200 versions of the same song with slight variations in it, but it's more or less the same song. If the model is trained with such a dataset, the result may turn out horrible such that it may recommend 50 copies that sound more or less the same. In such cases, deduplication becomes complex. Because if the dataset has 100,000,000 songs, you need to compare a song with every other song in the dataset. So it's a quadratic complexity of that problem, it's impossible to do from a time point of view in a single machine and you have to run it on hundreds and hundreds of machines.

Heterogeneity in data

Format, size and encoding techniques varies from data to data. A single dataset itself may have data in audio, video and text formats. If a dataset with only textual data is examined, some text will be in UTF-8, some in UTF-16, some in CSV, comma-separated format, some others in tab-separated format, some having HTML code in the actual text, some having an additional weird like placeholders embedded inside the actual national language text. So it is

required to invest a lot of time and effort in just transforming the text into a uniform format and coding for the data. All six cases studied here have this problem.

Data Preprocessing

Real-world data is often incomplete, inconsistent, and erroneous. So data preprocessing is an inevitable task before creating datasets which resolves the issues inherent with raw data. As data is coming from different sources, there can be missing data, wrong values, and ill formatted data which spoils the consistency and needs to be solved before feeding it to the DL models.

Dirty Data

Raw data comes up with a lot of imperfections like missing values, wrong values, and ill-formatted values. These unclean or noisy data is known as dirty data. Deep neural networks are good at deriving patterns from the given input. So, it is dangerous to feed noisy data to the DL models. Also, the DL experts might not be experts in the domain and so they are totally unaware of what needs to be filled when there are missing values and how to identify the wrong or ill-formatted values. For example, if there is a column for age and some of the values are missing. The system is supposed to make predictions based on each individual user and you do not have the age for 10% of them. That column can be filled out with the average or minus one. In order to fill the column, it is required to know what the column is meant to be and what can be filled in to replace the missing/wrong/mis-formatted values. All practitioners agree that they have faced this unclean data issue in all the cases they handled until now and in most of the cases, discussion with the people who collected the data was the only practical solution.

Managing categorical data

Categorical data are nothing but variables with label values instead of numerical values. Categorical variables can be both nominal as well as ordinal. Deep learning models cannot operate on label values as it requires all input variables in the numeric form. Even though one-hot encoding is used very frequently, it can be frustrating during implementation. When there are thousands of categories, the complexity again increases. If you have text data, for example,

that needs to be cleaned up and transformed into numeric form. Then it might not be possible to do it with a laptop or even a big server. There are core systems like Hadoop, Spark or Google DataFlow where big data processing can be done. However, it's still very dependent on the person doing it, what they are comfortable with, and also the data, how big is it, how difficult is it and what needs to be done with it. There are no predefined sets or standards to handle this.

Managing sequences in data

Metadata management should be considered with equal importance in order to manage the sequences in data. Storing the sequencing data alongside the contextual metadata is a bit challenging especially when the data quantity is too large. For instance, for chronological data, there is a time series which needs to be divided chronologically somehow, so you do not end up predicting the past.

Dataset Preparation

During dataset preparation, the main large dataset is divided into three different sets namely training, validation and testing dataset. Deep understanding of domain and problem will aid in relevant structuring values in the datasets.

Data Leakage

Data leakage is the challenge of not splitting the training and validation/test dataset properly so that the training data for the model happens to have the data which needs to be predicted. For instance, data leakage happens when the same data instance occurs in both training and testing dataset. This hides the actual performance of the model and when it is exposed to new and unseen data, the performance will not be as expected. So proper attention should be taken while splitting the dataset. Based on the study, we could infer that checking the data distribution is not always a solution to reduce data dependency.

Data Quality

Quality of data is crucial as poor quality data can cause severe performance degradation and exaggerated results. Data consistency is one of the factors deciding the quality of data. However, consistency is a hard to achieve target in many applications. For example, based on our study, the images collected from the hospitals are all taken in different conditions with different lighting. Accuracy, completeness, timeliness, validity are some other factors that ensure the quality of data. However, there is no exhaustive list of factors that should be checked to ensure the quality of data which is challenging.

Data Testing

Testing the data is a critical step which ensures the data quality and reduces the possible occurrence of defected data that affects the efficiency of the process. Absent, obsolete or wrong test data may prevent the practitioners from executing the test cases or give unreliable test results.

Expensive Testing

Data testing is highly expensive in the sense that it requires a lot of effort and time to define and automate test-cases specific to DL models. It's pretty hard to do regression testing on data, because data comes from users out in the wild where exerting control is impossible.

Tooling

Tooling comes as a challenge in most of the phases of the data pipeline. The major advantage of conventional software systems is that there exists a large variety of tools, especially for testing. As deep learning is a recently emerged approach, tools for testing such models are yet to be developed. All the cases included in our study experience tooling problem.

Deployment

Dl models need to be operationalised or put into production to measure the real performance and to generate a positive return for the investment in system development. When systems are ready for deploying in production, there are a unique set of challenges encountered which is explained as follows.

Data Extraction methods

Training-serving skew is a typical problem encountered when running deep learning models in production where the data seen at serving time differs in some way from the data used to train the model, leading to reduced prediction quality. For example, Google once built a system called quick access in Google drive which recommends a list of documents to open[13]. When the system was built, they first extracted data and made a training dataset, trained a model on it and did the evaluation which looked great. So, they put it in production, and it didn't work. When analysed, they realised that when they extracted the data for the training, they had a certain pipeline that it went through, but when they put it in production they had the data extracted from an API, and that API wasn't matching with the extraction they had for training. So it was some additional transformation happening in the API that caused the model to not work.

Overfitting

Overfitting is the situation when deep neural network memorises and fits itself so closely to the training set that it loses the capability to generalise and make predictions for new and unseen data. For instance, in a medical imaging case referred above where tabular data is used along with images, it turned out that the model was just learning the ID number of a certain hospital, and that hospital was a popular hospital to which the more severe cases were sent. So actually, the model was not learning anything from the images, rather it was just learning that the patients in that hospital are more likely to be sick, which is because they were sent there.

Post Deployment

Continuous monitoring is required even after deploying the DL components in production. This is because the real-world data is prone to all kind of shifts and distribution changes and the model learns constantly. The possible data management challenges after deployment are listed below.

Changes in data sources and distribution

When a certain problem is modelled, a distribution is postulated based on the data available at that time. However, consistency in data distribution cannot be expected all the time. Consider the house price prediction system in our study which is trained on historical real-estate data. When some sudden environmental disaster or society-wide effect takes place, the usual distribution will be disturbed and the trend in data changes. When data distribution changes, deep neural networks may not always be able to handle the new distribution. A sudden change in the source that supplies data can also lead to unexpected and undesired outcomes.

Data Drifts

Data drifts are also known as data shifts which happen over time. When data shifts happen, deep learning models may deliver weird and erroneous results. Consider systems, such as mobile interactions, sensor logs, and web clickstreams. Whenever the business tweaks or updates happen, the data those type of systems generate changes continuously. The sum of these changes is data drift. Other common examples of structural drift are fields being added, deleted and re-ordered, or the type of field being changed. For example, to support a growing customer base, a bank adds leading characters to its text-based account numbers. This kind of data changes causes the bank's customer service system to conflate data related to bank account 00-56789 with account 01-56789. All practitioners agree that most of the cases that they handle are subjected to this challenge.

Feedback loops

Feedback loops are sometimes beneficial and at times detrimental. For instance, if you implement recommendation systems, of course, there will be feedback loops. Because, the data collected will be mostly from your own customers and if you give them suggestions on what to buy, of course, they will buy more of that. Then the model sort of reinforces itself.

During the case study, all the practitioners agreed that while building any deep neural network, management of data requires more effort and time than model creation and coding as there exists a number of readily available algorithms for performing any deep learning task.

Table 4.2: Mapping between data management challenges and use cases

Phase	Challenge	Use cases of DL components					
		RS[1]	WPP[2]	HPP[3]	MD[4]	FFD[5]	MS[6]
Data Collection	Lack of metadata	X	X	X	X	X	X
	Data Granularity		X	X			
	Shortage of diverse samples		X	X	X	X	
	Need for sharing and tracking techniques	X	X	X	X	X	
	Data Storage	X					
Data Exploration	Statistical Understanding		X			X	X
	Deduplication Complexity	X	X	X	X	X	X
	Heterogeneity in data	X	X	X	X	X	
Data Preprocessing	Dirty data	X	X	X	X	X	X
	Managing sequences in data					X	X
	Managing categorical data				X	X	
Dataset Preparation	Data Dependency	X	X	X	X	X	X
	Data Quality	X	X	X	X	X	X
Data Testing	Tooling	X	X	X	X	X	X
	Expensive Testing	X			X		X
Deployment	Data Extraction Methods	X	X	X	X	X	X
	Overfitting				X	X	
Post Deployment	Data sources and Distribution	X	X	X			
	Data drifts	X	X	X			
	Feedback loops	X					

[1]Recommender System [2]Wind Power Prediction
[3]House Price Prediction [4]Melanoma Detection
[5]Financial Fraud Detection [6]Manufacturing Systems

If companies are able to act quickly to embrace naive ideas and opportunities, they will gain a valuable first-mover advantage. Companies who can get their data management and DL capabilities in order now will be in prime position to benefit from the next generation of AI operations tools as soon as they hit the market. This could give them an opportunity to secure a decisive edge over the competition.

4.5 Conclusion

Deep learning has established itself as one of the most popular techniques in the area of Artificial Intelligence and data management is an integral part of deep learning models as the performance of these models largely rely on data. However, without extensive research and highly developed supporting infrastructure, companies may face significant challenges while building production-ready systems with DL components.

In this paper we identified main data management challenges while building systems with DL components. Six use cases were described to identify

the challenges and also exemplify the potential for making use of the AI and specifically the DL technique. For these cases, the main problematic areas and challenges with building these systems were identified. To clarify these problem areas in more detail, a set of 20 challenges were identified and described across the phases of data pipeline. The challenges identified in this paper help practitioners to foresee the roadblocks that may encounter while managing data for deep learning systems. It also provides an overview of research challenges to be addressed by the academic community. The study helps to identify the probable blind spots for the companies wishing to implement deep neural networks as well as guide future research.

CHAPTER 5

DataOps - Definition and Evolution

This chapter has earlier been published as
From Ad-Hoc Data Analytics to DataOps
Munappy, A. R., Mattos, D. I., Bosch, J., Olsson, H. H., Dakkak, A.
In Proceedings of the International Conference on Software and System Processes, (pp. 165-174), (2020, June), Association for Computing Machinery.

Data is the key asset for organizations as it helps in better decision making, analyze performance and solving problems, to analyze the consumer behavior and market and so on. Moreover, data is the backbone for many hot and trending technologies like machine learning and deep learning [15]. The increased importance of data leads to the acquisition and storage of data in higher volumes which in turn gave rise to fields like Big Data, data mining and data warehousing. Data being the fuel for the digital economy, the need for data products like machine learning datasets, dashboards and visualizations is tremendously increasing. Organizations invest in data science and data analytics to solve problems with the collected data. Organizations realize that data is the key factor of success and as a result, they invest an enormous amount of money in the development of data products [110]. Data products

are built through a sequence of steps called data life cycle wherein for each step there will be both hardware and software requirements. Consequently, it is very essential to find the right balance of investment in requirements in different stages of the data life cycle [110]. Data management, data life cycle management, data pipeline robustness, fast delivery of high-quality insights are some of the major data problems that prevent companies from achieving their full potential.

DevOps is a set of practices that helps to build a collaboration between software development and information technology operations which in turn reduces the software development lifecycle and helps in continuous and fast delivery of high-quality systems. Thus, it is a methodology adopted in Software Engineering to aid agile software development [111]. Agile methodology focuses on empowering individuals, rapid production of working software, close collaboration with customers and quick response to the change in customer requirements [112]. Agile development is directly facilitated by CI/CD practices because it aids in software changes reaching production more frequently and rapidly. Consequently, customers get more opportunities to experience and provide feedback on changes [113].

Industries apply agile methodology, DevOps and CI/CD methodologies in software development. Data being an artifact like code, data analytics can also be benefited by the application of best practices of these methodologies in data analytics. DataOps is a process-oriented methodology that is derived from DevOps, continuous integration/continuous delivery and agile methodology for the quick delivery of high-quality insights to the customers. Introduction of agile development, CI/CD methodologies, and DevOps paves way for collaborative working, faster fixes, increased team flexibility, agility, cross-skilled and self-improving teams.

Many companies have succeeded in implementing DevOps, agile and CI/CD practices in their organization. However, there are only a few companies that have succeeded in adopting DataOps practices. In order to advance the concept of agile development and CI/CD and move towards DataOps, there are several steps that need to be taken. These several steps taken by the companies form a stairway and contributes to the evolution model of DataOps. Although it resembles DevOps practices, applying the same practices in Data Analytics is quite challenging as both of these disciplines are unique in their own respect and the skill-set, interest of practitioners involved in Data Analytics are very

different from the people who are involved in Software development. Therefore, the challenges faced by companies at each stage of progression towards DataOps will be much different from challenges associated with the evolution of DevOps.

The contribution of this paper is three-fold. First, it analyses the various definitions of DataOps from the literature as well as from the interviewers and then derives a definition for DataOps including the main components identified. Second, based on a case study with a large mobile telecommunication organization, we analyze how multiple data analytic teams evolve their infrastructure and processes towards DataOps. Third, we create a stairway of the evolution process. DataOps is a recently coined term, it is important to understand how companies are progressing towards DataOps. The evolution model demonstrates the essential requirements to climb a step in the stairway and also lists the set of challenges encountered while moving from one stage to the next.

The rest of this paper is organized into six sections. Section II is a description of the background and related work. In section III, the research methodology adopted for conducting the study is introduced. Section IV focuses on the findings of the case study, framing the definition for DataOps and the evolution stages. Section V details threats to validity and finally section VI summarises our conclusions and completes this paper.

5.1 Related Works

The peer reviewed works related to DataOps are quite few in number. Ereth [48] in his paper discussed a working definition for DataOps. Sahoo et. al presented a study which compares DataOps to DevOps and outlined the DataOps process and platform as well as the data challenges in manufacturing and utilities industries.

According to Julian Ereth [48], DataOps is a collection of various practices and technologies, than a particular method or tool. His study has resemblance with the first part of this paper where a definition for DataOps is derived from the literature as well as from the practitioners' understanding. Using a multi-vocal literature review (MLR) approach supplemented by interviews, the author analyzed and derived a definition for an ambiguous concept "DataOps". The author has also developed a framework that differentiates

between the exploration of DataOps as a discipline, which includes methods, technologies and concrete implementations, and the investigation of the business value of DataOps. However, the paper does not discuss how DataOps is different from Big Data Analytics, DevOps or CI/CD approach.

P. R. Sahoo et. al defines DataOps as an application of DevOps to data and they draw a parallel between DataOps and DevOps concepts. The authors define DataOps as DevOps for data analytics which eliminates inefficiencies, creates opportunities for collaboration, and promotes reusability to reduce operational costs. The study highlights how DataOps can be used in the data analytics discipline to bring revolutionary changes to business [114]. Also, it identifies the six significant steps of the DataOps process such as business requirements planning, data acquisition, data transformation, data repository management, data modeling, and insight publication.

Previous studies on data-driven development [115] describe the way companies evolve through their ability to use data. The study shows that companies follow a predictable pattern and start with an ad-hoc and manual approach to a data-driven approach. The authors developed a stairway with evolution stages. The first stage is the ad-hoc data collection. Challenges with manual data collection lead to automated data collection, followed by the introduction of dashboards that automatically updates with data from the field. After this stage, due to the constant flow of new insights evolving dashboards are introduced. Eventually, data-driven decision making is adopted for everything including sales, performance reviews, hiring and other processes. This work has a close resemblance to our study as it deals with data and evolution phases.

5.2 Research Methodology

The goal of this study was to formulate a definition for DataOps and to identify the phases of DataOps evolution.

Setting the RQs

The RQs defined in the study are as given below:-

- **RQ1.** How do practitioners define “DataOps”?

- **RQ2.** What are the different maturity stages Ericsson has gone through while trying to evolve from ad-hoc data analysis to DataOps?

To set the basic understanding of DataOps concepts and the essential components, we adopted the Multi-Vocal Literature Review approach following the instructions given by [79]. Then we conducted an interpretive single-case study, following the guidelines by [65], to acquire a deeper understanding of the data analytic approach followed at Ericsson. The main focus of this study is to understand and explain how the DataOps approach is perceived by Data Scientists, Data Analysts and Data Engineers to shorten the end to end data analytic life-cycle time and to enable collaboration. The impediments identified at each phase are based on our interpretations of the experiences of experts who work with data in a real-time scenario with real-world data collected from edge devices. The multiple cases from different teams in the same company are used in this study because it facilitates the exploration of a particular concept in a real-life setting as well as through a variety of lenses [108]. The overall research design and major steps in the process of the study are described below.

Multi-Vocal Literature Review

An MLR is a form of a Systematic Literature Review (SLR), which includes the Grey literature in addition to the published literature (e.g., journal and conference papers) [116]. Grey literature in SE can be defined as any material about Software Engineering that is not formally peer-reviewed nor formally published. The multi-vocal literature review approach was selected because it allowed us to gain more understanding of DataOps practices. As explained in [79], we analyzed if there is a great potential for benefiting from grey literature in the DataOps study and we identified that clearly, this approach is the best-suited one for studying DataOps. Because, the formal literature on the other hand DataOps is highly limited and on the other hand, there are quite several blogs, video media, and technical reports. Moreover, MLRs are useful since they can provide summaries of both the state-of-the-art and practice in a given area. We searched the academic literature using the Google Scholar, IEEE Xplore, ACM digital library and the grey literature using the regular Google search engine.

Need for MLR:

To learn more about the concept of DataOps, we did an initial search for the formal academic literature in different databases such as Google Scholar, IEEE Explore, ACM digital library, Web of Science, Scopus and ScienceDirect. However, we could not find a considerable number of peer-reviewed papers on the topic. Consequently, we decided to conduct a Multivocal Literature Review, based on all available literature on a topic.

According to Ogawa et.al a broader view about a particular topic can be obtained by using this wide spectrum of literature as they include the voices and opinions of academics, practitioners, independent researchers, development firms, and others who have experience on the topic [117].

Garousi et al. state that the practitioners produce literature based on their experience, but most of them are not published as academic literature. Also, the voice of the practitioners better reflects the important current state-of-the-art practice in SE. Therefore, it is important to include Grey literature too in the systematic review [116].

Process of MLR

The Multi-vocal literature review procedure adopted for the study is demonstrated in Fig. 1. The systematic review employs a string-based database search to select relevant studies from the literature. All retrieved literature was exported to MS Excel for further processing. The exported references were screened based on inclusion-exclusion criteria. The inclusion and exclusion criteria considered in our study are as shown below.

- **Inclusion Criteria :**
  (1) Papers and Google links describing the steps of the DataOps approach, essential components of DataOps, benefits, and challenges.
  (2) Papers describing the Big data pipelines, Big data processing pipelines
- **Exclusion Criteria :**
  (1) Duplicates and non-English

Exploratory Case study

The study was conducted in collaboration with Ericsson. Ericsson is a Swedish multinational network and telecommunications company. The company pro-

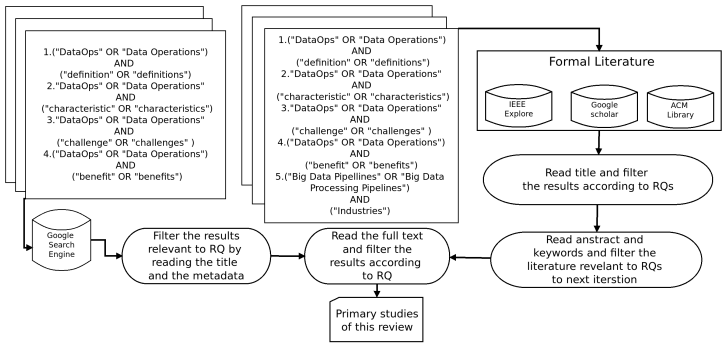

Figure 5.1: Multi-vocal literature review procedure applied in the study

vides services, software, and infrastructure in information and communications technology. The objective of the study is to explore the essential stages of the Data Analytic approach which Ericsson follows in their real-world settings and also to investigate its similarity to the popular DataOps approach. Each case in the study refers to a team at Ericsson working with the data they collect from different sources. For the study, a sample pool of Data Scientists, Data Analysts and Data Engineers were selected by one of the authors according to their expertise in the area of Data Analytics. Selected practitioners were invited to participate in the interview study and 4 of them showed interest to participate. After the interviews, interviewees were asked to suggest the names of their colleagues whom they think will be potentially interested in the study. Invitations were sent out to them as well and 4 of them participated in the study thus making a total of 8 interviews. Table 1 illustrates the role of our interviewees and the use cases.

Data Collection

Empirical data was collected through semi-structured interviews. Based on the objective of the research, to explore the data analytic approach employed at Ericsson, an interview guide with 45 questions categorized into six sections was formulated. The first and second sections concentrated on the background of the interviewee. The third and fourth sections focused on the data collection and processing in various use-cases and the last section inquired in detail about data testing and monitoring practices and the impediments faced dur-

Table 5.1: Description of use cases and roles of the interviewees

Case	Use cases at Ericsson	Interviewed Experts	
		ID	Role
A	Automated data collection for data analytics	R4	Senior Data Scientist
B	Building data pipelines	R1	Integration and Operations Professional
C	Toolkit for Network Analytics	R2	Analytics System Architect
D	Building CI pipelines for Data Scientist team	R7	Data Scientist
E	Tracking the Software Version	R5	Senior Customer Support Engineer
F	Testing the Software Quality	R6	Developer Customer Support
G	KPI Analysis Software	R3	Senior Data Engineer
H	Building data pipelines for CI and CD data	R8	Program Manager

ing every phase of the data pipeline. The interview guide was prepared by the first author and was reviewed by all the other authors. Based on the comments and recommendations some additional questions were added, a few similar questions were merged and some irrelevant questions were removed forming an interview protocol with 30 questions spread across six different categories. All interviews were conducted via video conferencing except for three which were done face-to-face and each interview lasted 50 to 100 minutes. All the interviews were recorded with the permission of respondents and were transcribed later for analysis.

One of the authors of this paper is an Ericsson employee who works quite a lot with the data teams. The first two authors of this paper are consultants at Ericsson and attend weekly meetings with Data Scientists and Data Analysts. Data collected through the meetings and discussions are also incorporated. The contact points at Ericsson were also a great help while validating the collected data.

Data Analysis

After the interviews, audio recordings of the interview were sent for transcription and a summary of each interview was prepared by the first author highlighting the important focus points of the interview. The investigated points from the summary were cross-checked several times with the audio recordings and interview transcripts obtained after transcription. A theoretical thematic data analysis approach was selected for coding [109]. The first author coded each relevant segment of the interview transcript in NVivo. For

the first iteration, the objective was to identify the use-cases discussed by each interviewee and phases of data analytics used by their team. After identifying the phases, a second iteration was performed to investigate the impediments encountered to completely set up DataOps practices at Ericsson. Thematic coding was performed by setting high-level themes as (i) Data Collection, (ii) Data Analytics, (iii) DevOps, (iv) Automation, (v) Data Testing, (vi) Data monitoring, (vii) Agile development. After careful analysis of the collected data, the first two authors agreed on the presentation of results in the paper. From the analysis, results were tabulated and sent to the other authors for collecting their reflections and then the final summary of the cases and results were sent to the interviewees for validation.

5.3 Findings

This section presents a definition of DataOps derived from literature as well as from the definitions given by experts during the interview study. Based on the study, we have constructed a five-stage evolution model of data strategy adopted at Ericsson to meet the evolving requirements of the customer. Our study is carried out with eight use cases as mentioned above. Requirements for moving from one step to the next and impediments encountered at each phase are identified and described as following.

Definition of DataOps

The exploratory case study and interviews show that different practitioners have different understandings about DataOps. During the interview, practitioners defined DataOps as "a process which fills the gap between data and operations team", "an efficient way of managing the activities in the entire data life cycle", "a method to showcase the interdependence of end to end data analytic process" or "an approach to eliminate data silos by connecting different data pipelines“.

Similarly, there are several definitions for DataOps in the grey literature. The concept of DataOps was first introduced by Lenny Liebmann in his blog post titled “3 reasons why DataOps is essential for big data success.” in 2014 [118]. However, it got popularity in 2015 through the blog post “From DevOps to DataOps” [119] by Andy Palmer. Andy Palmer described DataOps

as a discipline that “addresses the needs of data professionals on the modern internet and inside the modern enterprise” [119]. Gartner’s glossary defines DataOps as "hub for collecting and distributing data, with a mandate to provide controlled access to systems of record for customer and marketing performance data, while protecting privacy, usage restrictions and data integrity" [120]. There are several other definitions like DataOps "spans the entire analytic process, from data acquisition to insight delivery“ [121], "is a better way to develop and deliver analytics" [122], “is a new way of managing data that promotes communication between, and integration of, formerly siloed data, teams, and systems” [123] or "is illustrated as intersecting Value and Innovation Pipelines" [124]. From the above definitions, it can be seen that many of the experts define DataOps as an end-to-end process spanning from data acquisition to the insight delivery.

Many of the authors and interviewees emphasized the terms collaboration, automation, orchestration, integration and so on while expounding their definition of DataOps. For instance, "For DataOps to be effective, it must manage collaboration and innovation" [124], "DataOps is an analytic development method that emphasizes communication, collaboration, integration, automation, measurement and cooperation between data scientists, analysts, data/ETL (extract, transform, load) engineers, information technology (IT), and quality assurance/governance" [121], "Collaboration is the main part of both DevOps and DataOps" [50]. When describing their understanding of DataOps, most of the experts and authors elaborate their definitions with DataOps components and set of goals. Data pipelines to better explain the flow of data through operations [124], [114], [121], [125], the process of orchestration and automation. After analyzing the definitions, it was found that different definitions of DataOps seem to take different perspectives. While some focus on the activities of DataOps some focus on the goals of DataOps. Some focus on the technologies involved while some focus on the organizing structure of teams and so on. Tables 2 and 3 below categorizes the definitions we analyzed from interview studies and literature respectively.

From tables 2 and 3, it can be observed that some elements are common in all the definitions. Another important insight is that many terms associated with DataOps are also common to DevOps, Agile development and Big Data Analytics. We are also trying to identify the components/factors which make DataOps different from the others. We analyze the principles, goals, tooling,

Table 5.2: Analysis of DataOps definitions from literature

Perspective	Definition
activities	DataOps is not the product. It is an enabler of success [110]
activities	enables data analytics teams to thrive in the on-demand economy [126]
activities	help data teams evolve from a an environment with data silos, backlogs, and endless quality control issues to an agile, automated, and accelerated data supply chain that continuously improves and delivers value to the business [125].
way of working	DataOps is more than DevOps for data analytics because the deployment of a data pipeline is not a use case by itself [127]
way of working	focus on improving the communication, integration and automation of data flows between data managers and consumers across an organization [120]
way of working	works on data Management practices and processes which improves the accuracy of analytics, speed and automation [128]
way of working	DataOps uses technology to automate data delivery with the appropriate levels of security, quality and metadata to improve the use and value of data in a dynamic environment [129]
goal	The goal of DataOps is to create predictable delivery and change management of data, data models and related artifacts [130]
goal	bring rigor, reuse, and automation to the development of data pipelines and applications [125]
goal	By adopting DataOps, organizations can deliver data products in a consistent, reliable, fast, scalable, and repeatable process just like a factory [110]

Table 5.3: Analysis of DataOps definitions from interviews

Perspective	Definition
activity	a process which fills the gap between data and operations team
goal	approach to eliminate data silos by connecting different data pipelines
goal	method to showcase the interdependence of end to end data analytic process
goal	reduce the risk of poor data quality and exposure of sensitive data that may cause problems for the organization
activities	enables a continuous and dissipated flow of access to and insights from data
activities	automate the build of pipeline environments and give data pipeline developers self-serve ability to create, test, and deploy changes
way of working	intersection of advanced data governance and analytics delivery practices that constitutes the data life cycle.
way of working	is a way of avoiding common mistakes organizations make in data science and analytics
way of working	connects data creators with data consumers to increase collaboration and digital innovation.
way of working	an efficient way of managing the activities in the entire data life cycle
way of working	brings together the suppliers and consumers of data thereby escaping from a static data lifecycle

and people involved in all these different approaches/practices and formulate a definition for DataOps.

Definition for DataOps:

> *"DataOps can be defined as an approach that accelerates the delivery of high-quality results by automation and orchestration of data life cycle stages. DataOps adopts the best practices, processes, tools and technologies from Agile software engineering and DevOps for governing analytics development, optimizing code verification, building and delivering new analytics thereby promoting the culture of collaboration and continuous improvement."*

Even though DataOps has similarities with DevOps, agile methodology and big data analytics, it is still different from these existing approaches. It is a process-oriented approach to data that spans from the origin of ideas to the creation of graphs and charts which creates value.

DevOps merged Development and Operations teams to promote continuous integration and continuous delivery. Similarly, DataOps merges two data pipelines namely value pipeline and innovation pipeline. Value pipeline is a series of activities that produces value or insights and innovation pipeline is the process through which new analytic ideas are introduced in the value pipeline. In DevOps, the focus is on code and in data analytics, the focus should be both on code and data at every step. Moreover, DataOps has to deal with people along with tools due to which it requires a combination of collaboration and innovation.

DataOps and Big Data analytics are the two terms used interchangeably. However, DataOps is not only for Big data, instead it can be applied to any size of data to improve quality, speed and reliability of data insights.

In agile methodology, innovation happens in regular intervals. DataOps adopts this from Agile and as a result data team publishes new or updated analytics which is pushed into the value pipeline. Instead of copying best features of different approaches, DataOps borrows the best practices, technologies and tools and hand tailor it so that it fits to the unique context of data analytics.

Our definition of DataOps mostly aligns with the definition formulated by Ereth in [48]. In our definition, we call DataOps as a data strategy, because it sets the basis for transformation. Data strategy is something that is required by all organizations who make use of data for analytical purposes.

Use cases at Ericsson

Representatives of teams using raw data for developing data analytics, data engineers and data scientists were interviewed. The sections below describes the activities that they perform at Ericsson.

Case A: Automated data collection for data analytics - Ericsson delivers software every second weekend to base stations located different parts of the world and data is collected from all of the base stations that are used on the continuous integration flow. Thus, several base stations run test cases 24/7. When the test case fails, it immediately generates some data, specifically test case related metadata and the log from the base station. This data is sent to the cluster where it is ingested, unzipped, packaged, and so on. The data thus collected is further utilized for performing software data analytics.

Case B: Building data pipelines - Data pipelines are built for easier production of insights from raw data which is collected from the devices. With the usage of data pipelines, the entire data process starting from origin of ideas to literal creation of charts can be done with a minimum human involvement. The execution of different stages of the pipeline can be controlled by the scheduler which triggers the execution of one job immediately after finishing the current one. To manage the evolving customer requirements, underlying code for the data pipelines are kept scalable. The data pipelines can be either same or different for different customers depending on the similarities in their requirements.

Case C: Toolkit for Network Analytics - Network analytics utilizes different types of network data collected from the devices out in the field to identify interesting and useful trends and patterns. This internal toolkit can monitor, analyse and troubleshoot networks automatically whenever an equipment fault is found. After the development of this toolkit, Engineers are able concentrate on high value tasks, consultant requirement got reduced and it shows a conservative saving of man-hours. This toolkit produces professional reports for the customers and enables new opportunities by providing real-time and historical data. Whenever the schema of the input data changes, then the pipelines will not take it and this scenario requires human intervention.

Case D: Building CI pipelines for Data Scientist teams - The targeted customers for this case are data scientist teams who make use of hardware analytics for predicting the quality of the hardware delivered to the customers. When the customers sent their product to the screening centers or

the repair centers of Ericsson, the data gets recorded. The data scientist teams are collecting data from these centres to develop hardware analytics. The results or insights produced from the data can be used for machine learning algorithms for different activities. For instance, to predict if the customer is going to return the product or when the customer is going to return the product. This use case deals with building continuous integration pipeline for this data scientist team so that they get the feedback data continuously from the customers which can reduce the time for doing analytics as the data scientist team can get the data continuously. Apart from that CI pipeline will have basic unit tests and data linting tests.

Case E: Tracking the software version - To shorten the cycle time towards the customers there should be feedback loop from the customers. However, that's been very difficult, as the customers are in other countries and different companies. In order get data back from the customers software version running at the customer site needs to be tracked. Every third week software is delivered to the customers and then follow up is done to check which software they're unning, and also collect some performance data to ensure that the networks are performing adequately. Apart from the data at customer side, there is also data from internal CI environments which requires follow up. If an issue occurs in a lower level testing context it can be seen in high level testing or vice versa. So, the fingerprint of a certain issue can be seen across all the test levels. It's quite important to relate these issues. Otherwise there can have a bug which appears with ten different symptoms in ten different test environments, and it's difficult to debug. And the third is the customer data. The customers typically have a very good knowledge of their networks, they're very skilled at analytics, also. But, sales departments might not have same skills. Thus, it is required to help those departments understand data by creating dashboards out of data.

Case F: Testing the software quality - Features of the deployed software and those that are planned to be released in the future releases needs to undergo software quality tests with the help of KPIs. KPIs formulated will check if the system introduced on the software are reflecting what is expected as per design. This applies for the upgrades in the features as well. Data collected from the customers like counters are used to formulate KPIs. KPIs are used for monitoring if the feature behavior is as expected or as designed. If the KPIs are following the usual trend, then the performance is as expected.

There are two different tools which help in KPI monitoring.

Case G: KPI analysis Software - KPI analysis software helps to turn the KPI analysis into informed business decisions. KPI analysis is performed on the nodes in the continuous deployment zone before and after product updation. There is a mechanism to collect data automatically from the nodes. After getting the data, KPIs are defined manually and given to the software which then learns the trends of the counters or the KPIs and calculates it in the baseline. Once the updation happens, it again collects data from the devices and does the same again, but this time it compares the newly learnt trend with the baseline and the result of this will be charts or graphs representing performance improvements or performance degradation. These insights are delivered to the customers in an agile fashion. i.e every third week. After deploying the software, the team continuously monitor the data from the nodes.

Case H: Building data pipelines for CI and CD data - Building data pipelines for Continuous Integration and Continuous deployment enables access to data for all the team working with analytics. The main objective of this use case is to provide availability of high quality data to all the teams who are using data. A data pipeline with 4 steps such as data ingestion, data downloading, data archiving, data processing and data serving is built and it is manually monitored continuously to check for the data quality and data availability. Whenever there is a variation from the usual pattern, immediately the person responsible for the pipeline robustness is informed and that person finds the reason for the error and fixes it. So, for this particular use case, monitoring part involves human intervention as most of the tasks are done manually.

Evolution of DataOps

Based on the cross case analysis and literature, we identified a five stage evolution which happened before the introduction of DataOps. Because, the cases described above were not built in a single stretch to implement DataOps. Rather, they were built over time without even knowing that these would become beneficial in the future. From this inference, we thought of developing an evolution model with different stages that the company has gone through. Each of the cases mentioned above is developed at different stages. When climbing the stairway of evolution model, these cases/components are either

taken as such or necessary modifications are performed to take it further to the next stage. There might have other components as well in each of the stages. However, we are not considering all of those components. Instead, we consider those cases which are developed at some stage and taken over to the successive stages.

The stairway shown in fig 2 depicts the different maturity levels or evolution stages of data collection wherein data was initially collected in an ad-hoc fashion and progressing to completely autonomous unit that collects data, does data analytics, monitors itself for anomalies thereby reducing the time for delivering insights. The cases studied at Ericsson are mapped as components used at each of the evolution phases.

This evolution of stages in the taxonomy occurs on a component basis. Essentially, data life-cycle activities (*data collection, data preparation, data analysis, and delivering insights*) starting from origin of ideas to literal creation of values in the form of charts and graphs are performed at all maturity stages. The four phases of the "DataOps Evolution Model", namely "Ad-hoc data collection", "Data Pipelines and Data technologies", "Agile Data Science", "Continuous testing and Monitoring" and "DataOps", are described in detail in the remainder of this section.

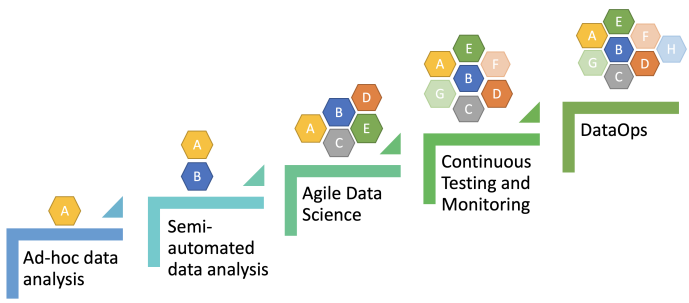

Figure 5.2: Evolution of DataOps

Phase 1: Ad-hoc Data Analysis

In ad-hoc data analysis, the reports or insights are created on-demand due to which the reports were highly customized. Usually, an ad-hoc analysis was performed to answer a very specific business question. The ad-hoc analysis was used in the early days of Business analytics. This was highly dependent on

the templates provided by the IT department. An ad-hoc analysis lets the user decide which data sources to fetch from and the way of data presentation. Ad-hoc insights can range from simple one-off data table all the way to intricately detailed sales reports using dashboards, interactive maps, and other advanced visualization features. Another important feature of ad-hoc data analysis was its ability to deal with different data sources in a flexible and scalable way. The ad-hoc analysis is helpful when there is a requirement of delivering immediate results. However, the reports generated out of this analysis are not used after the intended purpose. To make good business decisions, it is always better to have proper data engineering, data collection, and extensive data analysis. One of the practitioners commented that

> *"Because the data collection is basic plumbing, you know? You're moving one bit from one place to the other, you have to set up how the flow of data that goes from one end to the other. But, fully automated data analysis is something that we initially struggled with."*

Requirements for this phase:

To do ad-hoc data analytics, there should be some technology with which real-time data can be collected from multiple data sources.

Challenges:

All the data collected from different sources would not be in a single access point. Data silos at this phase prevent the customer from getting the full picture. Business decisions rely on a small amount of data which is not often sufficient to make decisions. Improper handling of an organization's data can lead to conflicts in the business values developed. When the underlying data varies throughout the organization, it can lead to conflicting results and delayed decisions.

Phase 2: Semi-Automated data analysis

Data pipelines for collecting and processing data are a much more efficient and automated way to implement data analytics. One of the interviewees said that

> *"It is more complex to build a robust data pipeline that is robust, reusable, scalable, secure and traceable in a real-world scenario. Data pipeline which we have built right now is reusable, secure and traceable, but we are not quite sure about its scalability and robustness."*

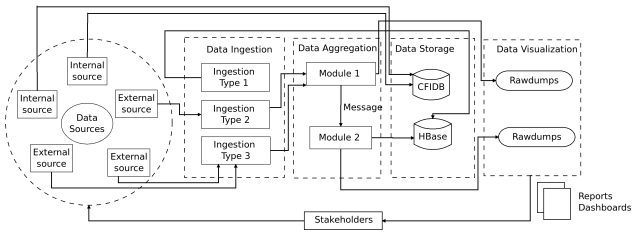

Figure 5.3: Big Data Analytics Pipeline at Ericsson

With the advent of data pipelines, data technologies and data processes became a necessity as they control and co-ordinate the different phases of data pipelines.

Data Pipelines: The huge volume of raw data is generated through various sources both internal and external to Ericsson. Data generated by different teams contribute to Internal sources and data generated by the devices at the base stations contributes to the external sources. Ericsson follows a similar data pipeline as explained in [131]. Data is collected in the form of raw dumps. Considering the complexity, heterogeneity, and volume of big data, Ericsson executes its applications in various stages as described below.

From figure 3, it can be understood that the data pipeline shown is a value pipeline as it creates visualization or insights from the collected raw data. According to the use cases, the data pipeline changes its face while the basic steps or structure being the same.

Data Technologies: Data pipelines require technologies to ingest, clean, analyze and visualize data. The technologies used to manage the pipeline can be categorized into four namely Data Engineering, Data Preparation, Data Storage and Data Visualization.

A. Data Engineering: The Data Engineering step performs two different operations at a high level, which include data collection and data ingestion. The process starts with continuous data streams collected from multiple sources including internal as well as external sources and ingested into the data pipeline. The data ingestion is important because data ingestion method itself is different for different data sources. For instance, the data ingestion method used for ingesting CI (Continuous Integration) data collected from internal sources is different from the ingestion method used for ingesting CD (Continuous Deployment) data collected from the external sources. Data ingestion is

capable to collect, import and process data from different data sources.

B. Data Preparation: Despite the collection of highly relevant data, analytics should take into account data heterogeneity to maintain efficiency in real-time applications. Data preparation involves the preparation of metadata links to the path where the actual data is stored and aggregating all the links for different types of data. Identification of encoded/encrypted data takes place here and once these kinds of encoded messages are identified, message to decode it is sent to the third-party servers. Encoded data are decoded by third-party servers and the respective metadata links are sent back to the aggregation module.

C. Data Storage: Metadata links prepared by the data preparation module is then stored in the Hadoop database. Teams can search for the metadata links in the database and can download the raw data dump files through the downloader. There are two different databases - CFIDB and Hadoop Database where the storage of data happens. CFIDB stores the CI data initially when it is collected from the internal teams at Ericsson. Hadoop database is the main database where the storage of aggregated data-logs takes place.

D. Data Visualization: After preparation and storage, the process of data analytics is executed. According to the requirement, different teams at Ericsson access the downloader to download the raw data dumps from the Hadoop database. After downloading the raw data dumps from the database, steps like data cleaning, data filtering, data processing, data transformation, etc are performed according to the requirements of the stakeholders. Most of the stakeholders require reports on the data showing the performance variation after the installation of a particular device.

Requirements for this phase: Well designed data pipelines are needed to efficiently evaluate, test, ingest, transform, validate, and publish data at scale. Data technologies for data collection, data engineering, data processing, data analysis, and data visualization. Also, data processes to control and coordinate data technologies as well as data pipelines are required

Challenges Lack of data pipeline robustness. Some of the activities are not automated. For instance, monitoring is done manually and whenever some issues are found, dependencies need to be contacted manually to fix the issue. The tickets raised while encountering problems in the data pipeline takes too long to get fixed. Once the insights are delivered, the process basically stops.

Feedback from the customers is not collected for further improvement.

Phase 3: Agile Data Science

Development and deployment are well defined by agile and DevOps methodologies. With these, teams are able to develop fully tested, functional code in a very short duration. Teams store their work in a common central repository in order to synchronize. There are a number of tools to aid the development and deployment phases. Customer requirements change with time and in order to cope up with the evolving requirements, it is required to follow the agile methodology in which insights are delivered in short sprints. Ericsson has a three weeks sprint means insights are delivered to the customers every third week, gets feedback from them and rework if required. A major challenge identified is that most of the time, stakeholders are not quite sure about their requirements. One of the interviewees quoted that

> *"A lot of the time you might create a dashboard or service which the stakeholders think they want, but at the end of the day, we see that it's almost never used. It usually takes some back and forth before we're able to find that killer app for the stakeholder."*

Requirements for this phase: Continuous delivery of business values to the customers. Evolving requirements from customers should be addressed. Customers should be delivered their demands frequently. Data team and customers should interact and the customers should communicate their requirements directly to the data team. Team should adjust themselves to increase the efficiency after regular intervals

Challenges Without continuous automated testing, a lot of man-hours are required to guard the flow in the data pipelines. To deliver insights quicker, good quality data should be made available. For instance, if the data source is not sending data, it should be detected as early as possible so that actions can be taken immediately. For this, data flowing through the pipelines should be monitored continuously.

Phase 4: Continuous testing and monitoring :

Continuous testing and monitoring of the data pipeline is an essential element while dealing with real-time data. Because it can help to detect the problems immediately before it is carried over to the successive stages of the pipeline. Without a monitoring mechanism, when the data received at the end of the pipeline is not as expected, it will be hard to identify the reason for the unexpected output. Also to meet the quality constraint, it is very important

to have automated unit tests as well as higher-level testing. Unavailability of higher-level testing was quoted as a challenge by one of the interviewers and it goes like this

> *"We are deploying automatically, I think, in most cases, but we don't have the quality checks and balances that we need to have. So, you can push something which doesn't have adequate quality and the pipeline still accepts it."*

With automated alerts, the concerned team can be notified when something goes wrong with any of the stages in the pipeline or if the pipeline is broken and the team can take proper measures to ameliorate the effect of the breakage.

Requirements for this phase: Test cases for testing the quality of data flowing through the pipeline. Automated mechanism to perform continuous monitoring and automatic alerting mechanism to send alarm to the responsible team when encountered with pipeline issues. Mitigation strategies should be developed in order to handle pipeline breakage

Challenges When there are pipelines, there should be some way to manage and orchestrate operational characteristics of the pipeline. Mechanism to push new data analytic ideas into the existing value pipeline

Phase 5: DataOps

DataOps shortens the end-to-end cycle time of data analytics, from the ideation phase to the insight development. As data lifecycle has dependency on people in addition to tools, it incorporates Agile Development practices into data analytics according to the organization's requirement thereby bringing the data consumers and data suppliers work together more efficiently and effectively. DataOps also adopts DevOps principles to effectively manage the artifacts like data, metadata and code. One major difference between DevOps for data analytics and DevOps for software development is that former has to manage both data and code whereas latter concerns only about the evolving code. With DevOps, it brings the two foundational technologies - continuous integration and continuous delivery which are two essential factors contributing to the goals of DataOps [124], [125], [132].

In addition to the DevOps lifecycle, data lifecycle has got an intersection between two pipelines namely value pipelines and innovation pipelines. Value pipelines are used for the creation of insights and innovation pipeline is for injecting the new analytic ideas into the value pipelines. In DataOps beyond the automated deployment of infrastructure, software, and application code, there

is the requirement of orchestration. The data pipeline starting from from acquisition of raw data to development of data product typically but not always follows a directed acyclic graph. DAG data structure has nodes and edges connecting the nodes. Nodes are the tasks where data is stored and edges denotes the flow of data from one node to another. The edges are directed because data cannot flow in the opposite direction. The output of one task becomes the input for another. The DAG is always acyclic because moving from node to node will never create an edge to a previous node. As the execution of steps occurs in a specific sequential order respecting the dependencies between different components, DAG usually requires orchestration. However, with the rise in real-time streaming architectures choreographed DAGs are becoming more popular. Because of the above mentioned reasons, automation, Orchestration, collaboration are the most important elements of DataOps.

Ericsson wants to apply DataOps to accelerate the data analytics workflow. At this point, the organization is heading towards the last step of the evolution of the stairway, which is DataOps. At least there is an initiative to organize all the people who work on data as a team so that data silos can be reduced. Moreover, with this initiative, all the teams associated with data can get to know what the other team is doing which makes the whole process of data analytics better. DataOps requires this sort of reorganization of the teams along with the value pipeline and innovation pipeline. However, there are concerns regarding all the data teams downloading data from the same place. Because the existing pipeline might not be able to serve a larger number of data requests.

There are several value pipelines created according to the requirements from the customers. However, all these pipelines share a common skeleton. Although, there is innovation pipeline, it is not much established and it is hard to explain how the new analytic ideas are pushed into the value pipeline.

Requirements for this phase: Data pipelines for creating insights and innovation pipelines for pushing new analytics into data pipelines. Continuous integration and continuous delivery practices for data analytics. DevOps for Data analytics. Mechanism to monitor and control the entire data life cycle process. Orchestration and advanced automation and agile practices for data analytics

Challenges Organizational restructuring is required. Unavailability of skilled team proficient in both Data analytics and DevOps is another chal-

lenge. Lack of interest of data scientists in learning new tools and technologies and data silos are the other major impediments.

5.4 Threats to Validity

There are three categories of potential threats to the validity of our work. This include construct validity, reliability and external validity that needs to be taken into consideration. To ensure construct validity, a few cases were excluded from the results as some of the interviewers did not had proper understanding of DataOps. As a result of the screening process, our study have some limitation with number of interviews. However, this limitation can be counted as an opportunity for further inquiry in future works. For reducing the researcher bias, the interviews were conducted by two researchers. To minimize internal validity threats, one of the co-authors, who has in-depth knowledge about the data processes in the company, was asked to validate the findings. Also, the findings were validated with other employees at the company. External threats we foresee are how the findings can apply to other organizations. Moreover, our reliance on grey literature as data sources for analysis also serves as a limitation. Further validation can be done by involving more organizations, which we see as future work.

5.5 Conclusion

DataOps is becoming increasingly popular in the industry due its ability to accelerate the production of high quality data insights. This paper proposes an evolution model describing a strairway with five steps showing how DataOps was evolved. With our research contribution, which is based on an extensive case study at Ericsson, we aim to provide guidance on this topic and enable other companies to establish or scale their DataOps practices. Our main contribution is the "DataOps Evolution Model". In the model, we summarize the five phases of evolution and maps cases to the phases in which they are used. Researchers and practitioners can use this model to position other case companies and guide them to the next phase by suggesting the necessary features. As future research, we plan to validate our model with other companies.

CHAPTER 6

Data Pipeline Management: Challenges and Opportunities

This chapter has earlier been published as
Data Pipeline Management in Practice: Challenges and Opportunities
Munappy, A. R., Bosch, J., Olsson, H. H
In International Conference on Product-Focused Software Process Improvement (pp. 168-184). Springer, Cham.

Data is being increasingly used by industries for decision making, training machine learning(ML)/deep learning(DL) models, creating reports, and generating insights. Most of the organizations have already realized that big data is an essential factor for success and consequently, they use big data for business decisions [133] [134]. However, high-quality data is critical for excellent data products [135]. Companies relying on data for making decisions should be able to collect, store, and process high-quality data. Collecting data from multiple assorted sources to producing useful insights is challenging [136]. Moreover, big data is difficult to configure, deploy, and manage due to its volume, velocity, and variety [137].

The complex chain of interconnected activities or processes from data generation through data reception constitutes a data pipeline. In other words, data pipelines are the connected chain of processes where the output of one or more processes becomes an input for another [138]. It is a piece of software that removes many manual steps from the workflow and permits a streamlined, automated flow of data from one node to another. Moreover, it automates the operations involved in the selection, extraction, transformation, aggregation, validation, and loading of data for further analysis and visualization [139]. It offers end to end speed by removing errors and resisting bottlenecks or delay. Data pipelines can process multiple streams of data simultaneously [140].

Data pipelines can handle batch data and intermittent data as streaming data [140]. Therefore, any data source will be compatible with the data pipeline. Furthermore, there is no strict restriction on the data destination. It does not require data storage like a data warehouse or data lake to be the end destination. It can route data through a different application like visualization or machine learning or deep learning model.

Data pipelines in production should run iteratively for a longer duration due to which it has to manage process and performance monitoring, validation, fault detection, and mitigation. Data flow can be precarious, because there are several things that can go wrong during the transportation of data from one node to another: data can become corrupted, it can cause latency, or data sources may overlap and/or generate duplicates [141]. These problems increase in scale and impact as the number of data sources multiplies and complexity of the requirements grows.

Therefore, data pipeline creation, management, and maintenance is a complicated task which demands a considerable amount of time and effort. Most of the companies do this maintenance manually by appointing a dedicated person to guard the data flow through the pipeline. This study aims to investigate the opportunities and challenges practitioners experience after the implementation of the data pipeline at their organization.

The contribution of this paper is three-fold. First, it identifies the key challenges associated with data pipeline management. Second, it describes the opportunities of having a dedicated data pipeline. These challenges and opportunities are validated through a multi-case study with three leading companies in telecommunication and automobile domains. Furthermore, the paper provides a taxonomy of data pipeline challenges including infrastruc-

tural, organizational, and technical ones.

The remainder of this paper is organized as follows. In the next section, we present the background of the study. Section III discusses the research methodology adopted for conducting the study. Section IV introduces the use cases and section V describes the opportunities created by the pipelines. Section VI details the challenges faced by practitioners while managing data pipelines. Section VII outlines the threats to validity. Section VIII summarizes our study and the conclusions.

6.1 Background

Several recent studies have recognized the importance of data pipelines. Raman et. al [138] describes Big Data Pipelines as a mechanism to decompose complex analyses of large data sets into a series of simpler tasks, with independently tuned components for each task. Moreover, large scale companies like Google, Amazon, LinkedIn, and Facebook have recognized the importance of pipelines for their daily activities. Data errors and their impact on machine learning models are described in [142] by Caveness et. al. They also propose a data validation framework that validates the data flowing through the machine learning pipeline.

Chen et. al describes the real-time data processing pipeline at Facebook [143] that handles hundreds of Gigabytes per second across hundreds of data pipelines. The authors also identify five important design decisions that affect their ease of use, performance, fault tolerance, scalability, and correctness and also demonstrate how these design decisions satisfy multiple use cases on Facebook. LinkedIn also has a similar real-time data processing pipeline described by Goodhope et. al in [144]. Data management challenges of deep learning is discussed by Munappy et. al through a multiple case study conducted with five different companies and classifies the challenges according to the data pipeline phases [145]. Lambda architecture proposed by N. Marz et. al and Kappa architecture [146] solves the challenge of handling real-time data streams [140]. Kappa architecture that considers both online and offline data as online is a simplified version of lambda.

Most of these studies illustrate the significance of data pipelines and the opportunities it can bring to the organizations. However, the challenges encountered in the industrial level during the development and maintenance of

Table 6.1: Outline of use cases and roles of the interviewees

Company	Use cases	Interviewed Experts	
		ID	Role
A	Data Collection Pipeline	R1	Senior Data Scientist
A	Data Governance Pipeline	R2	Data Scientist
		R3	Analytics System Architect
		R4	Software Developer
A	Data Pipeline for Machine learning Applications	R5	Data Scientist
		R6	Senior Data Scientist
		R7	Software Developer
		R8	Senior Data Scientist
B	Data Collection Pipeline	R9	Senior Data Engineer
		R10	Data Engineer
		R11	Data Engineer
		R12	Data Analyst and Superuser
C	Data Quality Monitoring Pipeline	R13	Director of data analytics team
		R14	ETL developer
		R15	Software Developer
		R16	Product Owner for data analytics team

the data pipelines in production is still not completely solved.

6.2 Research Methodology

The objective of this study is to understand the existing data pipeline as well as the challenges experienced at the three case companies and to explore the opportunities of implementing a data pipeline. Specifically, this study aims to answer the following research question:
RQ: What are the practical opportunities and challenges associated with the implementation and maintenance of Data Pipelines at the industry level?

Exploratory Case Study

A qualitative approach was chosen for the case study as it allows the researchers to explore, study, and understand the real-world cases in its context in more depth [147]. Since the concept of data pipelines is a less explored topic in research, we have adopted a case study approach [65]. Moreover, the case study approach can investigate contemporary real-life situations and can provide a foundation for the application of ideas and extension of methods. Each case in the study pertains to a use case that makes use of data. Table 6.1 details the selected five use cases from three companies.

Data Collection

Qualitative data was collected by means of interviews and meetings [61]. Based on the objectives of the research, to explore and study the applications consuming data in the companies, an interview guide with 43 questions categorized into nine sections was formulated. The first and second sections focused on the background of the interviewee. The third and fourth sections focused on the data collection and processing in various use-cases and the last section inquired about data testing and monitoring practices and the impediments encountered during the implementation and maintenance of data pipelines. All interviews were conducted virtually via videoconferencing due to the COVID-19 pandemic. Each interview lasted 40 to 60 minutes. The interviews were recorded with the permission of respondents and were transcribed later for analysis. The first author is an action researcher for the past one year/six months at company A and B respectively who attend weekly meetings with data scientists and data analysts. The data collected through these means are also incorporated.

Data Analysis

The contact points at the companies helped with analyzing the parts of the pipeline as well as the infrastructure used for building that pipeline. These notes together with the codes from transcripts were further analyzed to obtain an end-to-end view of different use cases. The audio transcripts were investigated for relations, similarities, and dissimilarities. The interview transcripts and meeting notes were open coded following the guidelines by P. Burnard [148]. After careful analysis of collected data and based on the inputs from the other two authors, the first author who is an action researcher at two of the companies developed the findings of the study which were then validated with the interviewees from the companies by conducting a follow-up meeting. For further validation and to collect feedback from a different team, the findings were also presented before another panel including super users, managers, software developers, data engineers, and data scientists at all three companies who were not involved in the interviews. The results were updated according to the comments at each stage of the validation which in turn helped to reduce the researcher bias.

6.3 Use cases

In this multi-case study, we explore data pipelines in real-world settings at large-scale software intensive organizations. Company A is within the telecommunication industry with nearly 100,000 employees who distributes easy to use, adoptable, and scalable services that enables connectivity. Further, we investigate Company B from automobile domain with 80,000 employees manufacturing its own cars responsible for collecting data from multiple manufacturing units as well as repair centers. Company C with 2,000 employees focus on automotive engineering and depends on Company B and does modular development, advanced virtual engineering and software development for them. In this section, we present five use cases of data pipelines studied from these three case companies A, B and C.

Case A1: Data Collection Pipeline

The company collects network performance data(every 15 minutes) as well as configuration management data(every 24 hours) in the form of data logs from multiple sources distributed across the globe which is a challenging activity. Data collection from devices located in another country or customer network requires compliance with legal agreement. The collected data can have sensitive information like use details which needs responsible attention. Furthermore, data generated by sources can be of different formats and frequencies. For instance, data generation can be continuous, intermittent or as batches. Consequently, the data collection pipeline should be adaptable with different intensities of data flow.

When data collection pipeline is implemented, these challenges should be carefully addressed. Fig. 6.1 shows the automatic data collection pipeline that collects data from distributed devices. In this scenario, the device is placed inside a piece of equipment owned by customers. However, the device data is extracted by filtering the customer's sensitive information. Base stations have data generation devices called nodes as well as a device for monitoring and managing the nodes. Data collection agents at the customer premise can interact either with nodes directly. However, access service is used for authentication. The data thus collected is transmitted through a secure tunnel to the data collection toolkit located at the company premise which also has access service for authentication. Data collection toolkit received the data

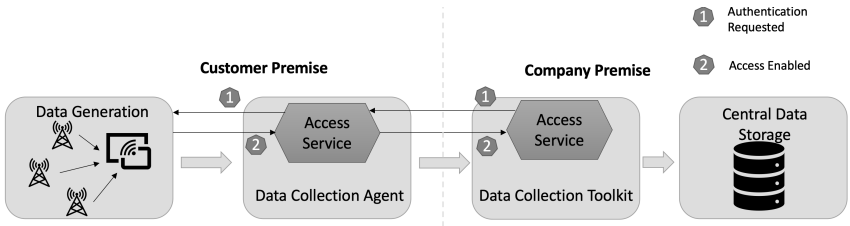

Figure 6.1: Data Collection Pipeline

and store it in the central data storage from where the teams can access the data using their data user credentials.

Case A2: Data Governance Pipeline

Fig. 6.2 illustrates the data pipeline that serves a subset of teams in the company who are working with data whenever they need it (With the term 'data', we mean the link from which the original data can be downloaded). This data pipeline gets two types of data dumps: internal and external which is the performance management data collected in every 15 minutes from the devices deployed in the network. The internal data dump is the data that is ingested by the teams inside the company and external data dump is the data collected directly from the devices in the fields. The data ingestion method varies according to the data source and the ingested data is stored in the data storage for further use. The data can be encrypted form which needs decryption before storing it. Data archiver module sends encrypted data dump to the third-party services for decryption. Decoded links from the third party are transferred to data storage. Therefore, data from distributed sources are made available in a central location. Teams can request data from any stage of the pipeline. The monitoring mechanism in the pipeline is manually carried out by the 'flow guardian' who is responsible for fixing the issues in the pipeline.

Case A3: Data Pipeline for Machine learning Applications

Data for this pipeline is obtained from the devices that are sent to the repair center. Data pipelines for machine learning applications has four main steps namely ingest, store, transform and aggregate. Data generated by the source

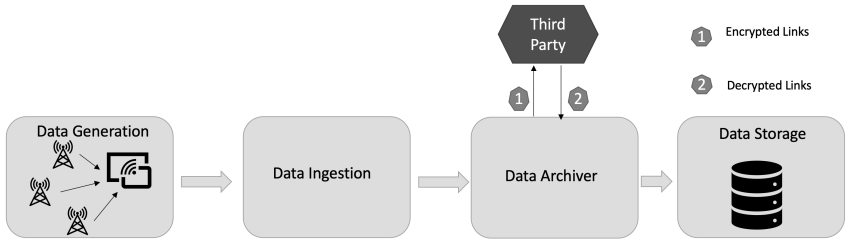

Figure 6.2: Data Governance Pipeline

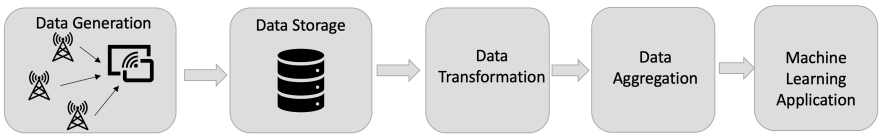

Figure 6.3: Data Pipeline for Machine Learning Applications

is gathered at a special zone in the field. The data ingestion module connected to those zones in the field collects data and ingest into the pipeline as batches. When new compressed files are found in the periodic checks, the transaction is logged and downloads it. These new files are then loaded into the archive directory of the data cluster. The data stored in the cluster cannot be used directly by the machine learning applications. Moreover, the data logs collected from different devices will be of different formats. Data transformation checks for the new files in the archive directory of the data cluster and when found, it is fetched, uncompressed and processed to convert it to an appropriate format. The converted data is then given as input to the data aggregation module where the data is aggregated and summarized to form structured data which is further given as input to the machine learning applications. Fig. 6.3 illustrates the data pipeline for machine learning applications

Case B1: Data Collection Pipeline

The Company B collects and stores three types of data and distributes it for teams as well as co-working organizations distributed around the globe. Plant data, delivery data, warranty data and repair data are the different

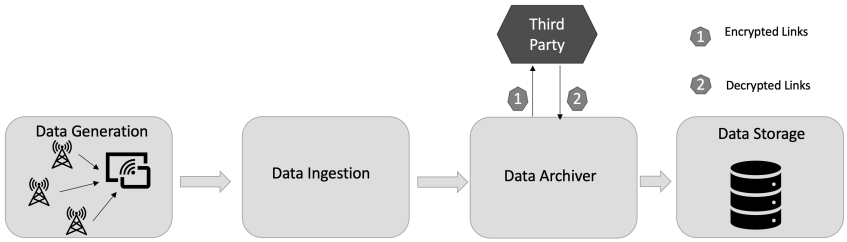

Figure 6.4: Data Collection Pipeline

types of data that are collected from sources such as manufacturing plants, service centers, delivery centers and warranty offices. The company B collects product data from distributed manufacturing plants every 24 hours. These manufacturing units will generate data for each product built there. However, not all the data generated by the plants are collected by the data collection agent of company B. Group Quality IT platform in the company demands the data that needs to be collected from the plants. Also, the data requested by the delivery centers are also collected and stored in the company's data warehouse. Fig. 6.4 illustrates the data collection pipeline working in company B. The data collected from different sources are in different formats and volume. Therefore, data transfer mechanism as well as data storage is different for all data sources. The data is ingested from the primary storage and then transformed into a uniform format and stored in a data warehouse which then acts as a supplier for teams as well as other organizations who demand for data. For instance, the delivery centers needs data about the products that are manufactured in the plants.

Case C1: Data Quality analysis Pipeline

The company C receives data collected and stored by company B and creates data quality reports which is used by data scientists team for analysing the product quality. For instance, the report can be used to understand the model that is sent to repair centers frequently. When the data quality is not satisfactory, investigation is initiated and actions are taken to fix the data quality issues. Company B sends data through private network to company C, and they store it in a data storage from where data scientists access it for

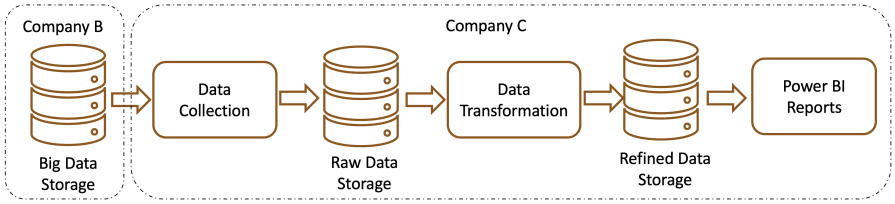

Figure 6.5: Data Quality Analysis Pipeline

creating reports and training machine learning models. Fig. 6.5 shows the data pipeline for data quality analysis at Company C.

6.4 Challenges to Data Pipeline Management

Based on our research, we see that organizations benefit from developing and maintaining data pipelines because of the automation it provides. On the other hand, there are certain challenges faced by practitioners while developing and managing data pipelines. This section describes the challenges of data pipelines derived through the interpretation of interviews based on the use cases described in section 4. After careful analysis of the challenges obtained from the interviews, we formulated a taxonomy for the classification of challenges namely Infrastructure Challenges, Organizational Challenges and Data Quality Challenges which are described in detail below.

Infrastructure Challenges

Data pipelines are developed to solve complex data infrastructure challenges. However, data pipeline management has to deal with some infrastructural challenges listed below.

Integrating new data sources: Data pipelines collect data from multiple distributed devices and make it available in a single access point thus solving data accessibility problem. However, the data sources increase rapidly in most of the business scenarios. Therefore, data pipelines should be able to integrate the new data source and also accommodate the data from that new source which is often difficult due to many reasons. Based on the empirical findings from the case study, three common reasons are listed below.

- The data source can be entirely different from the existing sources.
- Format of the data produced by the source might not be compatible with the data pipeline standards.
- Addition of the new source may introduce overhead on the data handling capability of the pipeline.

All the use cases except case C1 described in section 4 experience the challenge of integrating new data sources.

Data pipeline scalability: The ability of a data pipeline to scale with the increased amount of ingested data, while keeping the cost low is a real challenge experienced by the data pipeline developers. When the data produced by the source increases, the data pipeline loses the ability to transfer the data from one node to another leading to the data pipeline breakage and loss of data.

Increased number of nodes and connectors in upstream: Data pipelines are a chain of nodes performing activities connected through connectors that enable data transportation between the two nodes. Essentially, the nodes can have more than one capability. However, for the easy detection of faults, each of the nodes should be preferably assigned a single capability. Thus, the number of nodes and connectors increases in the upstream in relation to the data product yielded from the pipeline. This in turn increases the complexity of the data pipeline and decreases ease of implementation. The fragility and complexity of the data pipeline lead to inevitable delays in adding new types of activity data, which resulted in sticking new activities into inappropriate existing types to avoid human effort, or worse, not capturing activities at all. Practitioners R9, R10, R11 and R14 working on case B1 and C1 raised this challenge.

> "With the increased number of components in the data pipeline which in turn makes it difficult to understand and maintain. It is difficult to attain the right balance between robustness and complexity" - Senior Data Scientist (R6)

Trade-off between data pipeline complexity and robustness: To build a robust data pipeline, we should have two essential components called fault detection and mitigation strategies. Fault detection identifies faults at each of the data pipeline stages and mitigation strategies help to reduce the impact

of the fault. Including these two components increases the complexity of data pipelines. Moreover, it requires the data pipeline developers to anticipate the faults that can occur at each stage and define mitigation actions such that the data flow through the pipeline is not hampered. Some of the common faults can be anticipated and mitigated. However, it is not possible to identify all possible faults and define mitigation actions for those. Senior data scientists working on Case B1 and C1 and data scientist, R5 working on case A3 pointed out this as an important challenge.

Repeated alarms: Sending alarms are the most common and simple mitigation actions automatically taken by the data pipelines. Some faults take time to get fixed and during this time, the person or the team responsible for fixing the issues will get repeated alarms for the same issue. In the worst scenario, this can even lead to new alarms left unnoticed. Sending alarms is a mechanism adopted by all five data pipelines described in section 4. However, data engineers and software developers who participated in the study want to have an alternate mitigation strategy to replace the repeated automatic alarms in the data pipeline such as sending the notification only once and then waiting for a fix for some time.

> "Sending notifications is less appreciated by the teams as we get totally submerged in alarms during some days and some notifications are repeatedly sent and it is hard to identify new ones from the huge pile" - Senior Data Engineer (R9)

Organizational Challenges

This section gives a brief overview of the organization level challenges to data pipeline management.

Dependency on other organizations: Data pipelines can be spread between more than one company like case IV and V. Therefore, co-operation and collaboration are required from all the participating companies to maintain a healthy data pipeline. In most cases, external companies will have very minimal knowledge of what is happening in the other part of the pipeline. For instance, to deliver high-quality data product, company C requires support from company B as they are the suppliers of data.

Lack of communication between teams: Data pipelines are meant to share data between various teams in the organization. However, each team

builds pipelines for their use case and thus at least some initial activities are repeated in several data pipelines leading to redundant storage of data. Moreover, if any of the steps fails, the responsible person gets a notification from different teams at the same time for the same issue. Cases A1, A2, and A3 are collecting the same data and storing it in their databases. Data pipeline in case A3 can fetch data stored by data pipeline A2 instead of collecting raw data from the data sources. However, practitioners working on these use cases were completely unaware of these repeated activities in their pipelines.

Increased responsibilities of Data Pipeline owner: All faults in the data pipeline cannot be fixed automatically. Some faults demand either partial or complete human intervention. Therefore, a flow guardian or data pipeline owner is assigned for each of the pipelines who pays attention to data pipeline activities and takes care of the faults requiring a manual fix. Further, it is hard to assess what code might break due to any given change in data. With the increased use of data pipelines, the responsibilities of the flow guardian or data pipeline owner also increase. Practitioner R11 is assigned responsibilities of a flow guardian, and he has to manually monitor the data pipelines and initiate an investigation and fix whenever a problem is encountered. As Company C is also dependent on Company B, responsibilities are shared between R10 and R11. R10 takes care of request from Company C and R11 attends to the problems with company B.

> "Nobody wants to take up the responsibility of flow guardian. We feel that it consumes a lot of time and effort" - Director of Data Analytics Team (R13)

DataOps-DevOps Collaboration: When seeking to obtain better results from machine learning models require better, more focused data, better labeling, and the use of different attributes. It also means that data scientists and data engineers need to be part of the software development process. DataOps is concerned with a set of practices for the development of software and management of data respectively. Both concepts emphasize communication and collaboration between various teams of the same organization. DataOps combines DevOps with data scientists and data engineers to support development. The challenge of managing and delivering massive volumes of discordant data to those who can use it to generate value is proving extremely hard. Moreover, people working with data are less interested in learning new technologies and tools while it is not a hassle for DevOps users.

Data Quality Challenges

This section gives a detailed list of the data quality challenges due to improper data pipeline management.

Missing data files: Data files can be lost completely or partially during the transmission from one node to another. Fault detection mechanism can identify the exact point of disappearance. However, obtaining the missing files once again is a complicated task. Missing data files are only detected at the end of the data pipeline and in some cases, this results in poor quality data products. All the use cases experience the challenge of missing data files at different stages of data pipelines and one of the practitioners, R4 identified that 38,732 files had gone missing at a particular stage of the data pipeline over five months.

> "Data quality is a challenge that is being discussed over years. But, at industry level we still struggle to achieve desired level of data quality" - Senior Data Scientist(R1)

Operational errors: Data pipelines encounter operational errors which hampers the overall functioning. Operational errors are very common in non-automated data pipelines. Some parts of the data pipelines cannot be completely automated. Human errors at these steps are the reasons for operational errors. For instance, data labeling in a data pipeline cannot be automated completely due to the unavailability of automated annotation techniques that are compatible with all types of datasets. Practitioner R12, R13, R4, and R3 raised the problem of operational errors and their impact on their respective data pipelines.

Logical changes: Data drifts and change in data distribution results in the data pipeline failures due to the incompatible logic defined in the data pipeline. Therefore, the data pipeline needs to be monitored continuously for change in data distributions and data shifts. Besides, data pipelines should be updated frequently by changing the business logic according to the changes in data sources. Practitioner R12, R13, and R16 explained the struggles of working with outdated business logic in their data pipelines.

6.5 Opportunities

The previous section illustrated the challenges of data pipelines when implemented in real-world. However, there are many opportunities the data pipeline offers through automating fault detection and mitigation. In this section, we survey some of the most promising opportunities of data pipelines and how practitioners working on data are benefited by the implementation of it.

Solve data accessibility challenges

Data generated by assorted multiple devices are collected, aggregated, and stored in central storage by data pipelines without human intervention. As a result, data teams within and outside the organization can access data from that central storage if they have proper data access permissions. Accessing data from devices located on the customer premises is a difficult and tedious task. Most often, the devices will be distributed around the globe and teams has to prepare legal agreements complying with the rules of that specific country where the device is located for accessing data. When the data is stored after aggregation, data loses its granularity, and as a result, teams working with fine-grained data has to collect data separately. With data pipelines, teams can access data from any point of the data pipeline if they have necessary permissions. This eliminates repeated collection and storage of the same data by multiple teams.

Save time and effort of human resources

Automation of data-related activities is maximized through the implementation of data pipelines thereby reducing the human intervention. When a data pipeline has inbuilt monitoring capability, faults will be automatically detected and alarms will be raised. This reduces the effort of data pipeline managers and flow guardians. As the data pipeline is completely automated, requests by teams will be answered quickly. For instance, if the data quality is not satisfactory to the data analyst, he can request the data from the desired store in the data pipeline, and he receives it without delay. On the other hand, if the workflow is not automated, the data analyst has to investigate and find out where the error has occurred and then inform the responsible person to send the data again which eventually delays the entire data analysis process.

Moreover, the effort of the data analyst is also wasted while investigating the source of data error.

> *"We spent time cleaning the data to meet the required quality so that it can be used for further processing such as training or analytics. With the data pipeline, it is easy to acquire better quality data."* - Analytics System Architect(R3)

Improves traceability of data workflow

Data workflow consists of several interconnected processes that make it complex. Consequently, it is difficult to detect the exact point that induced error. For instance, if the end-user realizes that part of the data is missing, it might be lost during data transmission, while storing the data in a particular schema or due to unavailability of an intermediate process. The end-user has to guess all the different possibilities of data loss and has to investigate all the possibilities to recover the lost data. This is a time-consuming task especially when the data workflow is long and complex. Company C has reported that they have experienced this problem several times and as they are getting data from company B, it took a lot of time for them to rectify the error, and sometimes they won't be able to recover the data. After implementing data pipelines, the process of detecting faults is automated thereby increasing traceability.

> *"Everyone in the organization is aware of the steps and with data pipelines, you will have full traceability of when the pipeline slowing down, leaking, or stops working."* - Data Scientist(R5)

Supports heterogeneous data sources

Data pipelines can handle multiple assorted data sources. Data ingestion is a process through which data from multiple sources are made available to the data pipeline in a uniform format. Data Ingestion is the process of streaming-in massive amounts of data in our system, from several external sources, for running analytics and other operations required by the business.

> *"Data streams in through several sources into the system at different speeds and sizes. Data ingestion unifies this data and decreases our workload. Data ingestion can be performed as batches or real-time."* - Data Engineer(R10)

Accelerates Data life cycle activities

The data pipeline encompasses the data life cycle activities from collection to refining; from storage to analysis. It covers the entire data moving process, from where the data is collected, such as on an edge device, where and how it is moved, such as through data streams or batch-processing, and where the data is moved to, such as a data lake or application. Activities involved in the data pipeline are automatically executed in a predefined order and consequently, human involvement is minimized. As the activities are triggered by themselves, the data pipeline accelerates the data life cycle process. Moreover, most of the data pipeline activities are automated thereby increasing the speed of data life cycle process and productivity.

Standardize the Data Workflow

The activities in a data workflow and their execution order are defined by a data pipeline which gives the employees in the organization an overall view of the entire data management process. Thus, it enables better communication and collaboration between various teams in the organization. Further, data pipelines reduce the burden on IT teams thereby reducing support and maintenance costs as well. Standardization through data pipelines also enables monitoring for known issues and quick troubleshooting of common problems.

> *"Data pipelines provide a bird's eye view of the end to end data workflow. Besides, it also ensures a short resolution time for frequently occurred problems."* - Product Owner(R16)

Improved Data Analytics and Machine Learning Models

Organizations can make use of carefully designed data pipelines for the preparation of high quality, well-structured, and reliable datasets for analytics and also for developing machine learning as well as deep learning models. Besides, data pipelines automate the movement, aggregation, transformation, and storage of data from multiple assorted sources. Machine learning models are highly sensitive to the input training data. Therefore, quality of training data is very important. Data pipelines are traceable since the stages are predefined yielding better quality data for the models. Moreover, data pipelines ensure a smooth flow of data unless it fails in one of the steps.

Data Sharing between teams

Data pipelines enable easy data sharing between teams. Practitioners R4, R8, and R9 mentioned that the data collected from devices in the field are undergoing the same processing for different use cases. For instance, data cleaning is an activity performed by all the teams before feeding the data to ML/DL models. Therefore, there is a possibility of the same data going through the same sequence of steps within different teams of the same organization. Further, data storage also is wasted in such cases due to redundant storage. With the implementation of data pipelines, the teams can request data from a particular step in some other data pipeline and can process the subsequent steps in their data pipeline. However, the data pipeline should be able to serve the requests in such cases.

Critical Element for DataOps

DataOps is a process-oriented approach on data that spans from the origin of ideas to the creation of graphs and charts which creates value. It merges two data pipelines namely value pipeline and innovation pipeline. Value pipeline is a series of stages that produce value or insights and innovation pipeline is the process through which new analytic ideas are introduced into the value pipeline. Therefore, data pipelines are critical elements for DataOps together with Agile data science, continuous integration, and continuous delivery practices.

6.6 Threats to Validity

External validity: The presented work is derived from the cases studied with teams in the domains of automobile and telecommunication. Some parts of the work can be seen in parts of the company differently. All the terminologies used in the company are normalized and the implementation details are explained with necessary level of abstraction [149]. We do not claim that the opportunities and challenges will be exactly the same for industries from a different discipline.
Internal Validity: To address internal validity threat, the findings were validated with other teams in the company who were not involved in the study. Further validation can be done by involving more companies, which we see as

future work [65].

6.7 Related Works

This section presents the most related previous studies on data pipeline development and maintenance.

P. O'Donovan et. al describes an information system model that provides a scalable and fault tolerant big data pipeline for integrating, processing and analysing industrial equipment data [150]. The authors explain the challenges such as development of infrastructures to support real-time smart communication, cultivation of multidisciplinary workforces and next-generation IT departments. However, the study is solely based on a smart manufacturing domain. A survey study by C.L.Philip Chen et. al discusses about Big Data, Big Data applications, Big Data opportunities and challenges, as well as the state-of-the-art techniques and technologies to deal with the Big Data problems [151]. A Big Data platform Quarry is proposed by P. Jovanovic et. al [139] manages the complete data integration lifecycle in the context of complex Big Data settings, specifically focusing on the variety of data coming from numerous external data sources. Data quality challenges and standards/frameworks to assess data quality are discussed in many works [152] [141] [153]. Although there exists significant number of data quality assessment and mitigation platforms, the industrial practitioners experience data quality issues which indicates that the problem is not solved.

6.8 Conclusions

The multi-case study indicates challenges and opportunities involved in implementing and managing data pipelines. The challenges are categorized into three namely infrastructural, organizational, and data quality challenges. Nevertheless, the benefits data pipeline brings to the data-driven organizations are not frivolous. A data pipeline is a critical element that can also support a DataOps culture in the organizations. The factors inhibiting Data pipeline adoption were mostly concerned with human aspects e.g. lack of communication and resistance to change; and technical aspects e.g. the complexity of development. Suitability of completely automated data pipelines might be questioned for certain domains and industry sectors, at least for

now. However, a completely automated data pipeline is beneficial for the domains that can adopt it. Frequent updates are advantageous, but the effects of short release cycles and other data pipeline practices need to be studied in detail. Understanding the effects on a larger scale could help in assessing the real value of data pipelines.

The purpose and contribution of this paper is to explore the real-time challenges of data pipelines and provide a taxonomy of the challenges. Secondly, it discusses the benefits of data pipelines while building data-intensive models. In future work, we intend to further extend the study with potential solutions to overcome the listed data pipeline challenges.

CHAPTER 7

Modelling Data Pipelines

This chapter has earlier been published as
Modelling Data Pipeline
Munappy A. R., Bosch, J., Olsson, H. H., Wang, T. J.
In 2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA) (pp. 13-20). IEEE.

Data is becoming increasingly popular in the industry due to the importance of data products such as APIs, dashboards, benchmarks and report creations. The role data plays in the decision-making process and the development of ML and DL models makes it even more important. Therefore, all processes associated with data ranging from data generation to data reception need to be monitored. Fault detection, reporting and mitigating the effect of faults are complex but inevitable while building efficient data products.

Data pipelines are complex chains of activities that manipulate data where the output of one component becomes the input to the other [154] thereby allowing smooth, automated flow of data from source to destination. A data pipeline starts with a data source that generates data and ends at a destination that receives the processed data. The ultimate destination of a data pipeline

need not be data storage. Instead it can be any application such as a visualization tool [155] [156], Machine Learning(ML) models [157] [158] or Deep Learning(DL) models [159] [160]. Components in the data pipeline are capable of automating processes involved in extracting, transforming, combining, validating, and loading data [161]. Data pipelines can process different types of data such as continuous, intermittent and batch data [144]. Moreover, data pipelines eliminate errors and accelerate the end-to-end data processes which in turn reduces the latency in the development of data products. Hence, the usage of data pipelines is an absolute necessity for all data-driven companies.

Although data pipelines have the potential to overcome data management challenges through automation, monitoring, fault detection, etc, modeling data pipelines for a use case demands an unreasonable amount of time and effort. We need to identify the activities which consume data, the output of each activity, order of execution, monitoring methods, intermediate storages, where to place the storage in the pipeline, data collection method, etc and they varies between companies. Thus, modeling a data pipeline is important as well as time-consuming.

This study addresses the above mentioned problems by proposing a conceptual model which is developed based on a multiple case study performed at a large-scale telecommunication company. The contribution of this paper is three-fold. First, it describes the challenges associated with data management and the usage of existing data pipelines. Second, a conceptual model of an end-to-end data pipeline is presented that can be used as a reference while building data pipelines for applications such as ML/DL models to incorporate automatic monitoring, fault detection, mitigation and alarming techniques. The conceptual model is validated through a case study with three leading companies from manufacturing, telecommunication and automobile domains. Furthermore, the paper maps the challenges that can be potentially solved by the usage of the proposed data pipeline model.

The remainder of this paper is organized as follows. In the next section, we present the background of the study. Section III discusses the research methodology adopted for conducting the study. Section IV introduces the use cases and section V describes the challenges using the existing pipelines. Section VI details the data pipeline meta-model. Section VII describes the conceptual model that we use as a basis for our analysis. A validation study is detailed in section VIII and section IX outlines the threats to validity. Section

X summarizes our study and the conclusions.

7.1 Background

Data-driven development has been adopted by the companies realizing the benefits that can be yield from data [162]. Technologies for data-driven development needs enormous amount of data for their processing. Consequently, collection, storage, and processing of copious amounts of data became a necessity [163]. With the increased amount of data, data management challenges also increased [164]. Data pipelines can be a potential solution to overcome at least some of these challenges.

Data pipelines are broadly classified into two categories namely ETL and ELT. ETL stands for Extract, Transform and Load whereas ELT stands for Extract Load and Transform. P. Vassiliadis has presented a survey on Extraction-Transformation-Loading (ETL) processes and tools. The study describes a standardized approach for the construction of conceptual and logical modeling tools for ETL processes [165]. Also, each component of the E-T-L triplet is analyzed separately for the activities happening inside each component, identified the real-time challenges associated with each of them and solutions to overcome those challenges are explained in detail.

In [166], the authors have proposed a UML based conceptual approach to model ETL processes. A group of UML concepts is utilized to represent the ETL processes such as data sources integration, transformation, key generation and so on. The authors claim that a UML based approach makes the model simple to understand and powerful. Tilmann and Hans have described a big data analytics pipeline with abstract stages of it [167]. They also discuss the necessity of ETL-like processes in benchmarking and has proposed and implemented a framework similar to ETL.

A machine learning pipeline is proposed by Amershi et. al in [15] based on a case study at Microsoft. It discusses the nine stages of ML workflow along with the best practices followed at Microsoft. The authors mention the importance of data in AI applications by illustrating the three data related stages in ML workflow.

Although these studies lay strong foundation, practitioners experience several data quality challenges and issues around data governance and security while dealing with real-time data. Our study aims to design a conceptual

Figure 7.1: Research Methodology

model that can act as a domain specific language for fault tolerant data pipelines.

7.2 Research Methodology

The objective of this study is to understand the existing data pipeline as well as the challenges experienced at the case company and to develop a conceptual model of the robust data pipeline. Based on the study objectives, we formulated the following research questions:

- **RQ1:** What are the challenges related to data and data pipeline management that practitioners in the case company experience?
- **RQ2:** What are the essential elements of a fault-tolerant, automated, traceable end-to-end data pipeline?

The research methodology adopted for the study is illustrated in fig. 1.

Exploratory Case Study

A qualitative approach was chosen for the case study as it allows the researchers to explore, study and understand the real-world cases in its context in more depth [147]. Since the concept of the data pipeline is a less explored topic in research, we have adopted a case study approach [65]. Each case in the study pertains to a use case that makes use of data. Although the cases in our study are different use cases from the same company, they utilize data for different activities and can be benefited from the data pipeline we develop. Three selected use-cases at the company are given in table 1.

Data Collection

Qualitative data was collected by means of interviews and meetings [61]. Based on the objective of the research, to explore and study the applications

Table 7.1: Description of use cases and roles of the interviewees

Case ID	Use cases at case company	Interviewed Experts	
		ID	Role
A	Data collection pipeline for data analytics	R1	Senior Data Scientist
		R2	Data Scientist
B	Building data pipelines for data governance	R3	Data Scientist
		R4	Analytics System Architect
		R5	Software Developer
C	Machine learning pipeline	R6	Data Scientist
		R7	Senior Data Scientist
		R8	Software Developer
		R9	Senior Data Scientist

consuming data in the company, an interview guide with 45 questions categorized into six sections was formulated. The first and second sections focused on the background of the interviewee. The third and fourth sections focused on the data collection and processing in various use-cases and the last section inquired about data testing and monitoring practices and the impediments faced during each step of the data pipeline. The interview guide was prepared by the first author and was reviewed by second and third authors. According to the recommendations, extra questions were added, a few similar and irrelevant questions were removed and some questions were modified. Finally, an interview protocol with 30 questions across six different categories was developed. All except three interviews were conducted via videoconferencing. Each interview lasted 50 to 100 minutes. The interviews were recorded with the permission of respondents and were transcribed later for analysis. One of the authors works in the case company and the first author is a consultant who attends weekly meetings with data scientists and data analysts. Data collected through the meetings and discussions are also incorporated.

Data Analysis

The audio recordings of interviews were transcribed and a summary of it was prepared by the first author. The transcripts were investigated for relations, similarities, and dissimilarities. The interview transcripts were open coded following the guidelines in [168]. The first author prepared notes during the meetings with the team and analyzed them further. The main contact point who is also an author helped with analyzing the parts of the pipeline as well as the infrastructure used for building that. These notes together with the codes from transcripts were further analyzed to obtain an end-to-end view of different use cases. It also helped to understand the parts common to all use

cases. After careful analysis of collected data and based on the inputs from the other two authors, the first author developed the first conceptual model which got refined through iterations.

Validation Study

The validation study is performed both internally and externally through qualitative interviews followed by feedback sessions. First, the conceptual model of the data pipeline was presented by the first author to the teams inside the case company. The reflections about the data pipeline, overall opinion, agreements and suggestions for improvement were collected from every practitioner. These reflections are considered as internal validation.

For external validation of our findings, two manufacturing companies were selected. An interview guide was prepared by the first author for validating the conceptual data pipeline model. The first author presented the model. Second and third authors conducted an interview followed by a discussion to collect data. The entire session was recorded for the first case and for the second one, the first author took notes. Thus, feedback from 20 practitioners was collected and recorded. The conceptual model was then modified to address some of the issues raised during the discussions. Also, the inputs from the practitioners are incorporated in the conceptual model. The remaining issues will be addressed in future works.

7.3 Use cases

This section introduces the existing data pipelines used in the telecommunication firm. Each of these use cases is separate and there is no interaction between those pipelines.

Data Collection Process

The company collects data from multiple sources distributed across the globe which is a challenging activity. When data is collected from a device located in another country or from the customer network, it should be according to the legal agreement. Also, sensitive information in the data should be handled responsibly. Furthermore, data collection should consider the fact that different data sources generate data in different frequencies and formats.

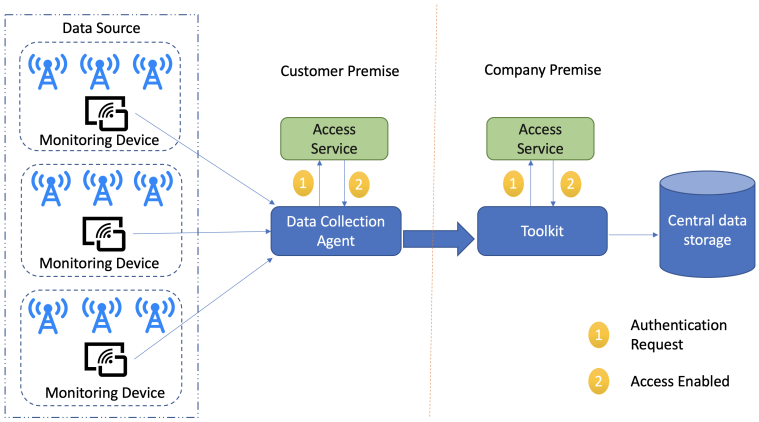

Figure 7.2: Data Collection Process

For instance, data can be collected continuously, intermittently or as batches. Moreover, the data collection mechanism itself should be capable to adjust with different intensities of data-flow.

When data collection is automated, these challenges should be addressed properly. Fig. 2 shows the automatic data collection from the devices. In this scenario, the device is placed inside a piece of equipment owned by customers. However, the device data is extracted excluding the customer's sensitive information. Base stations have got nodes as well as a device for monitoring and managing the nodes. Data collection agents are equipment located on the customer's premises (physical location) that can interact either with the nodes directly or with the device to collect the data. However, to ensure that the data collection agent has the right to collect data, it is authenticated with the help of access service. The data thus collected is transmitted through a secure tunnel to the toolkit located at the company premise. This data collection agent also needs the help of access service for authentication. Once the agent at the customer premise gets the data, it is stored in the central data storage. The teams can get the data from the central data storage.

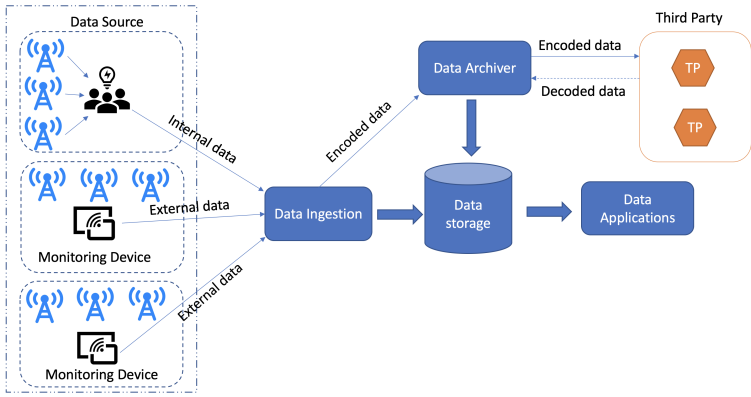

Figure 7.3: Data Pipeline for Data Governance

Pipeline for Data Governance

The data pipeline shown in fig. 3 is developed to serve the teams in the company who are working with data whenever they need it (With the term 'data', we mean the link from which the original data can be downloaded). This data pipeline gets two types of data dumps: internal and external. The internal data dump is the data that is ingested by the teams inside the company and external data dump is the data collected directly from the devices in the fields. The data ingestion method is different for different sources and the ingested data is stored in the data storage for further use. The data can have encrypted links that need to be decrypted before storing it. Whenever an encrypted data dump is found, the data archiver module sends it to third-party services for decryption. Decoded links from the third party are stored. Thus data from different sources are made available in a central location. Teams can request data from any stage of the pipeline. The pipeline is manually monitored by 'flow guardian' who is responsible for identifying the faults and solving them.

Pipeline for Machine Learning Systems

Data pipeline has four main steps namely ingest, store, transform and aggregate. Data is generated by the source which is gathered by a special zone in the field. The data ingestion module is connected to those zones in the field and the collected data is ingested into the pipeline as batches. When new

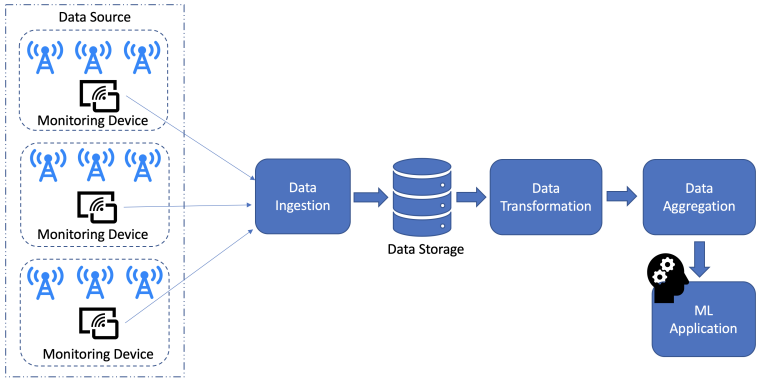

Figure 7.4: Pipeline for Machine Learning Systems

compressed files are found during periodic checks, the transaction is logged and downloads it. These new files are then loaded into the archive directory of the data cluster. The data stored in the cluster cannot be used directly by the ML applications. Moreover, the data logs collected from different devices can be of different formats. They need to be converted to a suitable format. This conversion is performed by the data transformation module. Data transformation checks for the new files in the archive directory of the data cluster and when found, it is fetched, uncompressed and processed to convert it to an appropriate format. The converted data is then given as input to the data aggregation module where the data is aggregated and summarized to form structured data which is further given as input to the ML models.

7.4 Challenges with data management

In this section, insights into data management challenges are presented based on the findings from our cross-case analysis. We have identified ten major challenges with data management and existing data pipelines used in the company.

Data Availability

The availability of the right data in the right format at the right time is a basic requirement for the successful development of data products. Data collection is a difficult task and sometimes it fails due to authentication failure, environmental factors or failure of collection device. Even after collecting an enormous amount of data from the devices, it may not reach the intended destination. Data collected can be incomplete. i.e not all information will be available in the data warehouse. For instance, due to software failures, parts of the data can be lost. Unless we have a tracking mechanism, this loss is hard to identify. Furthermore, if well defined data is given as input to the model, it won't be able to give the same performance when unseen real-world data is given leading to underfitting.

Data Quality

For data-hungry systems like ML and DL, data quality is crucial. When low-quality data is fed to the algorithms, the low-quality output will be produced. For instance, while collecting fault logs from the devices in the field, there should be a clear distinction between faults due to environmental factors and faults due to device failure. The challenge is when the data is transformed to fit a predefined structure, "unnecessary" parts of it are removed. Therefore, we would need a method to save the original file. When the data is transformed on the fly and then stored, this would not be possible. It is always good to have the original raw data file stored so that it can be accessed whenever the structured data becomes insufficient to meet the requirements.

Data-flow Instability

Data-flow to the company's data storage is not always stable. The device at the company's end should be prepared to receive the data from the distributed devices. The pipeline, if existing, should be capable enough to handle data-flow through it. If multiple teams request data simultaneously, the pipeline should be able to serve it. Moreover, elements in the pipeline should be monitored properly for the failures. Failure of pipeline elements lead to data-flow instability. Timely upload of processed data is mandatory especially in case of dependencies. If data pipeline is accepting continuous, intermittent and batch data, at some points, the inflow of data will be heavy and during

other times, it will have to handle only continuous data. This also lead to data-flow instability if the pipeline is not able to adjust its capacity.

Data Silos

A data silo is the gap between the data and the consumers who need the data. It is a result of poor architecture, legacy operational systems, and outdated company culture. The main problem is that the data becomes isolated and trapped without reaching the consumers. When individual teams develop their own data pipeline, it may also lead to data silos. There is a high probability that multiple teams performing the same activities for different use cases.

Data Dependencies

Data dependency occurs when a team or device has to depend on the outcome of some other team or device for starting their activity while developing any data product. There can have situations where the team has to wait for a long time for obtaining the required data. When the dependee is a software device that failed, the dependent won't be notified. Usually, in such cases after the expected time of delivery, necessary actions are taken to check the existence of dependee. Data dependencies often lead to delayed production.

Data Pipeline Latency and Overhead

Data pipelines can create additional latency to the entire data workflow. Latency is defined as the time taken by data to travel through the entire pipeline. When multiple teams are requesting data or when the data inflow increases, the data pipeline may produce delayed outputs. Besides that, data has to go through all components and checks in the pipeline to reach the destination. Failure or slow down any one of these components in the transit can lead to latency. The data pipeline becomes an overhead when the data is used for cases for which data quality is not important.

Data Pipeline Owner Overloaded

The data pipeline owner is a person who is responsible for monitoring and managing the data flowing through the data pipeline. During peak times,

that person gets overloaded. Moreover, it is always good to have maximum automation in the data pipeline.

Unreliable Data Pipelines

The reliability of a data pipeline depends on the reliability of its elements. Therefore, elements within a data pipeline should be made fault-tolerant. A data pipeline without built-in validation, mitigation mechanisms cannot ensure data quality.

Low Storage Capacity

When every team is constructing their own data pipeline, everyone stores the same data in different forms in the data storage leading to shortage of storage space. Databases, data warehouses, data lakes are for eliminating redundant storage of data. However, an increased number of data pipelines will eventually lead to reduced storage capacity. Another reason is the storage space division between teams. When storage is divided between teams, each one will get only a small portion of the actual available storage space.

7.5 Data Pipeline Meta-model

The section above described challenges with the data management encountered by the industry practitioners. From the empirical findings and through analysis of existing pipelines, we develop a conceptual model of data pipeline that can potentially overcome the limitations of existing data pipelines.

Meta-model is a set of concepts used to build data pipelines. Nodes and connectors are the two main basic components used to build a data pipeline. Nodes are interconnected with each other using connectors. Both of the components have certain capabilities. For instance, the ability to connect different nodes is the capability of the connector. Fig. 5 shows the components, capabilities of both nodes and connectors, notations used to represent nodes and connectors, etc. Colour coding, icons, and differences in style are used to represent the variations in nodes and data that is flowing through the nodes. Capability is the ability of a node to perform a certain activity. For instance, data sources in the pipeline have the capability to generate data.

Capabilities of Nodes: Data Generation, Data Collection, Data Ingestion, Data Storage, Data Processing, Data Labelling, Data Pre-processing, Data Reception.

Capabilities of Connectors: Data transmission, Authentication, Validation, Monitoring, Mitigation, Sending alarm.

The ability of the connectors to perform certain activity is termed as capability of connectors. Connectors have different capabilities. Connectors are the carriers of data. i.e they transmit data from one node to the next. Some connectors carry raw data and some carry processed data. Some connectors carry labeled data. Each of these connectors are given separate notation and color-coding in fig. 5. All connectors have the capability to transmit data. Apart from that, they have additional capabilities like authentication, validation, monitoring, mitigation and sending an alarm. The authentication capability of a connector is denoted by placing a light green color square with 'A' on top of it. Similarly, validation is represented by a yellow square with 'V' on it. Mitigation is represented by a red square with 'M'. Monitoring by the beige color square with 'F' on it. As the letter 'M' is already taken for mitigation, we use F(Fault detection). Grey color square with 'S' denotes the capability to send the alarm.

7.6 Conceptual Model of Data Pipelines

Data pipeline is a complex series of components interconnected with each other where the output of one component is fed as input to the other. The starting point of the data pipeline is called source and the final destination is called a sink. All other nodes are intermediate nodes. Each component in the data pipeline manipulates data by performing activities. The data pipeline model presented in this section is a conceptual model. According to the requirement, it can be used by any organization for any data application by creating instances. Fig. 6 illustrates the conceptual model of a data pipeline.

Data Generation: Data pipelines start from a source and in most real life cases the source will be multiple and distributed. Therefore, our pipeline also has multiple sources distributed all around the world. Data sources can be of different types. Any device having the ability to generate data is called a data source. Data sources considered here are classified into two categories: Human-generated and Machine generated. Data that is produced through

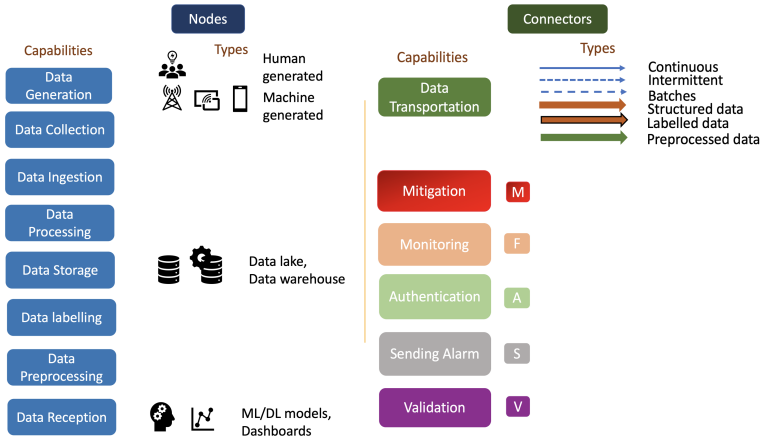

Figure 7.5: Meta-model for building data pipeline

manipulations by team members, other team members or other organizations come under human-generated data. Machine-generated data are produced by the various devices employed in different equipment, vehicles and so on. For instance, data generated by a device embedded inside the car, data produced by mobile applications, etc.

Data Collection: The data-flow from the source can be batch, intermittent or continuous. Three different styles of arrows starting from data sources indicate that the connector between the data source and data ingestion can carry all the three variations of data-flow. Although the data collection node can collect data from the sources, it should show the permission to collect data from the sources.

Raw Data Storage - Data Lake: The data collected from the source will be raw and should be stored so that the original data files can be retrieved in the future. However, the data collection node has to show its right to ingest data into the data pipeline. This authentication is carried out by connectors between data collection and data lake.

Data Ingestion: Raw data files can be taken from the data lake and ingested into the data processing component. This data ingestion method will be different for different types of data. Data can arrive in all shapes and sizes. Real-time stream data will be processed sequentially. The continuous data will be validated immediately after extracting it from the data lake.

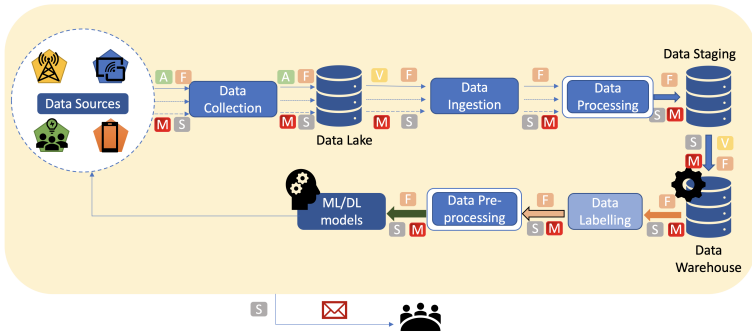

Figure 7.6: Conceptual model of data pipeline

Data Processing: Data processing itself is a composite node in which there can be multiple individual components like data aggregation, data parsing, data transformation, etc. Data aggregation is the process by which raw data is expressed in a suitable form for statistical analysis. With data transformation, the unstructured aggregate data is converted into a structured format or semi-structured format. Thus, the data processing step converts all different types of data into a single format and stored in the data staging area. This is symbolically represented in the fig. 6 with three different incoming arrows to data processing indicating continuous, intermittent and batch data. The output from the data processing step is a single thick arrow.

Data Staging and Data Warehouse: The data staging is a temporary storage area where the data can be stored for validation. After validating the structured or semi-structured data, it is stored in a data warehouse. This data warehouse functions as a point from which the data can be taken for several data applications like report creation, ML applications, dashboard creation, etc.

Data Labeling: Our study is mainly focused on ML applications. Therefore, the data pipeline shows the necessary steps for automating the data pipeline for ML applications. ML algorithms can be supervised, unsupervised or reinforcement. For unsupervised algorithms, the data labeling step can be skipped. That is the reason why the data labeling step is shown in a light blue color different from the other nodes. As most of the companies are using a supervised approach for their ML applications, the focus is more on the same.

Data Pre-processing: To achieve better performance from ML algo-

rithms, data needs to be pre-processed before training. The pre-processing depends on the practitioners developing the application and also the nature of the problem. Nevertheless, popular data pre-processing steps are data quality assessment, data imputation, data encoding, data sampling, dimensionality reduction, etc. Thus, data pre-processing can include any process which transforms data in such a way that it can be fed to a machine-learning algorithm.

Data Reception: The output from the data pre-processing is given as input to the ML models which uses it for training, retraining, testing, and validation. The ML model act as a data sink in fig. 6. As most of the companies are following agile methodology, the data will be collected back from the data products for further iterations. Therefore, data produced by these ML applications are again collected by the data collection node and goes through the pipeline continuously.

Capabilities of Connectors: Each connector between the nodes have the capability to send data to any other node, monitor the data-flow, fault detection, check for associated mitigation strategies when a fault is detected. If there is no defined mitigation strategy, then it has the capability to send alarms to the responsible team. The faults that may happen at each stage will be different. Therefore, mitigation strategies also will be different. Similarly, the responsible team/person who can handle a particular fault will be also different from each other.

To summarize, the conceptual model of the data pipeline is fully automated in which monitoring is performed throughout the pipeline. The data pipeline is fault-tolerant to some extent because mitigation strategies are there to ameliorate the effects of faults. Moreover, teams can request data from any point in the pipeline according to their requirement.

7.7 Validation Study

The conceptual data pipeline model was validated through interviews with 20 industry professionals from three companies of different domains and different maturity levels. Two of the authors, together with one author online conducted the interview study for validation. The purpose of the research was to develop a conceptual model of a fully automated, fault-tolerant and traceable data pipeline. Table 2 illustrates the challenges with data management that

can be partially or completely solved by the usage of data pipelines.

Data availability can be solved to a certain extent with the proposed data pipeline. However, when the source fails to generate data, then the data will not be available. The failure will be detected automatically and the corresponding mitigation strategies will be adopted. Also, when the customer is reluctant to share data, then, of course, the data will not reach the pipeline. In such a scenario, the corresponding manager will receive an alarm and they can take necessary actions to ameliorate the situation. Data quality challenges cannot be completely solved. If the data produced by the source is low quality, there is no inherent mechanism in the pipeline to make it high quality. However, high-quality data will not lose its quality during its transmission through the pipeline. Data-flow instability can be also solved using the proposed data pipeline with its built-in mechanisms to control the flow of data. Data silos is a complicated phenomenon that cannot be solved by the mere introduction of a data pipeline. It needs reorganizations, a cultural shift in the company, etc. Data dependencies can be completely solved as the pipeline itself is designed to be fully automated. Dependency between teams will be eliminated and all teams will have a dependency on the data pipeline. Data pipeline latency will be there. As the components in the pipeline are more and all those components have got intelligence in the form of capabilities, latency comes as a side effect. Data pipeline owner overhead can be reduced as the responsibility will be spread across multiple persons who have got a better understanding of a specific part of the data pipeline. The reliability of the data pipeline can be ensured with the connector level mitigation strategies. Failure of a particular component will affect the data-flow which will be detected by the monitoring mechanism and the fault is taken care of either by the mitigation strategies or by the corresponding responsible person. Storage capacity cannot be increased by implementing the data pipeline. As discussed earlier, it can eliminate the redundant storage of data. However, no provision in the data pipeline can increase storage capacity.

The model developed by the first author was presented before the industrial experts and their consensus and disagreements were recorded. The validation section will be structured in terms of agreements and suggestions for improvement. Agreements refer to situations where practitioners agree and confirm while suggestions for improvement refer to situations in which the interviewees had a different opinion. Some minor corrections were made to the model

Table 7.2: Analysis of data challenges that can be solved with the proposed data pipeline

Challenges with Data Management	Proposed Data Pipeline
Data Availability	Partially solve
Data Quality	Partially solve
Data-flow Instability	Completely solve
Data Silos	Cannot solve
Data Dependencies	Completely solve
Data Pipeline Latency	Cannot solve
Data Pipeline Owner Overloaded	Completely solve
Unreliable Data Pipeline	Completely solve
Low Storage Capacity	Cannot solve

and the other concerns raised are saved for extended work of the data pipeline model.

Case A: Manufacturing Company

The conceptual model was presented by the second author and the third author collected the feedback from the industry professional.

Agreements: The conceptual data pipeline model was identified as a standard concept that can be used by teams located in different parts of the world while building their data pipeline. With a standard architecture, there can have a common understanding of processes. Automation of monitoring and mitigation are interesting and important

Suggestions for Further Enhancements: Monitoring has three variations such as performance monitoring, data profile monitoring, and condition monitoring. Performance monitoring can continuously check for the software performances in the data pipeline. Condition monitoring can ensure the data pipeline health and data profile monitoring can make sure that the data flowing through the data pipeline meets the data quality requirements, detect faults and allows investigation of data problems thereby making it traceable.

Case B: Automobile Company

The conceptual model for data pipeline was presented by the first author and we collected the agreements and disagreements from all the industry professionals.

Agreements: Practitioners recognized this conceptual model as a language for communication between data professionals. When there is a common language, it is easy to avoid misinterpretations. Moreover, they all agreed about the monitoring spread across the pipeline and the need for different storage stages.

Suggestions for Further Enhancements: The major disagreement was concerning the nomenclature of nodes in the data pipeline. The data warehouse is a term to represent the aggregated and well-transformed data which is used for a specific use case. In the conceptual model, the data warehouse is capable of storing data for multiple applications. The suggestion was to rephrase all these ambiguous names such as data preparation and data processing. Another problem is that the model does not give attention to the reinforcement algorithm. Data pipeline does not explain who is the person responsible for each step of the pipeline. To increase readability and understandability for everyone in the company, it was suggested to have different model views. For instance, a model with no technical terms, a second one with much lesser abstraction and so on.

Case C: Telecommunication Company

Internal validation of the data pipeline is performed at the telecommunication company. The first author presented the work and recorded the responses from the team.

Agreements: The team liked the conceptual model of the data pipeline as they all were developing their pipeline for a particular use-case. They realized the need for a standard pipeline model that can be followed by everyone in the organization. Moreover, some of them were happy about storing original data in a data lake as they were experiencing issues with the availability of raw data files.

Suggestions for Further Enhancements: The data processing step itself can include data labeling and data preprocessing which is shown as separate steps in the proposed data pipeline model. They suggested it is good

to have separate storage at each step of the data pipeline. There should be a provision to stop sending alarms continuously to the team whenever the issue persists for a longer duration.

7.8 Threats to Validity

This study was based on existing data pipelines developed by different teams from the same company located in various parts of the world to reduce the bias of operating with a single team within the same organization. To address internal validity threat, one of the authors who has in-depth knowledge about the data process in the company, was asked to validate the findings. Also, the findings were validated with other teams in the company who were not involved in the study. Furthermore, the study was validated again by external companies from different domains.

7.9 Conclusion

In the immediate future, it will be inexorable that the daily analysis will not be able to keep up with the daily influx of data. Along with the increased popularity of data and its products, challenges associated with it also increased. Data scientists and other practitioners working with data spend a considerable amount of time combating with those challenges. The proposed data pipeline model can either solve or alleviate several data management challenges with limited human intervention. However, data pipelines need to be carefully designed so that data-flow can be monitored, managed and maintained. Therefore, fully automated, fault-tolerant and traceable data pipelines are gaining importance. The conceptual data pipeline model proposed in this paper has nodes and connectors which perform the activities in the data workflow. The conceptual model is validated using an exploratory case study where a total of 20 practitioners from three different companies participated. All of them agreed with the necessity of data pipelines in the organization, they liked the structuring of the conceptual model and the automation of monitoring, alarming and mitigation. They also gave a few suggestions for the improvement of the model. As future work, description on mitigation strategies at each step, the physical realization of the conceptual model will be done and the results will be analyzed.

CHAPTER 8

Fault Detection and Mitigation in Data Pipelines

This chapter has earlier been published as
Towards Automated Detection of Data Pipeline Faults
Munappy A. R., Bosch, J., Olsson, H. H., Wang, T. J.
2020 27th Asia-Pacific Software Engineering Conference (APSEC) (pp. 346-355). IEEE.

Data is becoming increasingly popular and is considered as the new oil in the current era. Essentially, the success stories of large scale companies like Google, Amazon, and Facebook triggers a data-driven culture in small-scale companies as well. Data quality is a critical factor deciding the success of data-driven organizations [169]. Detecting and repairing dirty data is one of the perennial challenges in data analytics, and failure to do so can result in inaccurate analytics and unreliable decisions [170]. Over the past few years, there has been a surge of interest from both industry and academia on data cleaning problems including new abstractions, interfaces, approaches for scalability, and statistical techniques [171] [170] [172]. To address this situation, the data pipeline is a pro-active solution that can be applied. A data pipeline is a piece of software that enables a smooth, automated flow of data from one

node to another and eliminates human involvement from the process.

Data pipelines that automate the steps ranging from data generation to data reception are seen as increasingly attractive by software development companies using data-intensive models. Moreover, a data pipeline is a progressive concept necessary to practice DataOps culture which is adopted by the companies to accelerate the data life cycle activities thereby accommodating evolving customer requirements and fluctuating market needs [173]. Data Pipelines decompose complex analysis of large data sets into a series of simpler tasks, with independently tuned components for each task. This modular setup allows the re-use of components across several different pipelines. However, the interaction of separately tuned pipeline elements produces poor end-to-end performance as errors introduced by one component cascade through the entire pipeline, significantly affecting overall accuracy. Therefore, while many software development companies have indeed succeeded in implementing data pipelines in parts of their organization, there are few examples of companies that have succeeded in implementing robust data pipelines that can perform continuous monitoring and failure recovery without human intervention [143]. To develop a fault-tolerant data pipeline, organizations need to identify the faults that may occur at each step and should define the corresponding mitigation actions that need to be taken. Anticipating the possible faults at each step and creating an exhaustive list is a non-trivial task. Further, defining mitigation strategies for each of the expected faults is arduous.

In this paper, we present the findings based on action research conducted at two companies in which we explore four data pipelines used for preparing data for machine learning models and also for report creation. While the continuous monitoring, automatic fault detection and definition of mitigation actions in data pipelines help in failure recovery and indeed minimizes the human intervention, it is not easy to anticipate all possible faults in various stages of the data pipelines. As a result of our study, we identify the typical faults that occur at different steps of data pipelines as well as the mitigation strategies adopted by the practitioners to ameliorate the impact of data pipeline failure.

The paper is structured as follows. First, we outline data quality, data management, and data pipelines to set the background of our study. Second, we describe our research approach based on action research involving three data pipelines from two companies located in Sweden. We present our findings in which we describe the faults encountered by the practitioners during the

development and maintenance of data pipelines and the possible mitigation strategies. In the following section, we present the evaluation process of the model by taking a small slice of the existing data pipeline and implementing fault detection and mitigation strategies. Finally, we present the works from industry and academia describing data pipelines followed by the conclusions.

8.1 Background

This section reviews some concepts on the importance of data management, data quality, and challenges faced by the companies. These concepts are complementary to each other and form the basis of this work.

Data Quality

Data quality is a critical factor that determines the success of any data product such as data analytic reports, machine learning models, or deep learning models. Data quality challenges are quintessential in domains such as health [174] [175], cyber-physical systems [176], embedded systems [177], etc. Data quality is lost due to several factors such as transmission errors, infrastructural problems, human errors, and system errors. To overcome these types of typical errors, an efficient, systematic, and automated data management system should be developed.

Data Management

The quality of data can be improved through efficient data management approaches. Thus, data management plays an important role in increasing productivity and profits, higher accuracy and reliable results, and better decisions. With the advent of machine learning and deep learning models, the importance of data management increased even more. Kumar et. al discusses the data management challenges in production machine learning which details a comprehensive review of various data analytic systems and analyzes key data management challenges and techniques [178]. A similar study on data lifecycle challenges in production machine learning is conducted by Polyzotis et. al in [27]. Data management challenges for deep learning are described in detail by Munappy et. al in [145]. These studies clearly indicate the interest from academia and industry in the area of data management.

Data Pipelines

Building data pipelines for data management is a method through which many manual steps can be eliminated from the process and enables a smooth, automated flow of data from one node to another [179]. Data pipelines allow the collection of data from multiple assorted devices distributed around the world. Data pipelines facilitate batch as well as stream processing. Moreover, a data pipeline can define the activities involved in the data lifecycle and their order of execution.

8.2 Research Methodology

In this paper, we report on an action research study conducted in close collaboration with two companies in the embedded systems domain [66]. The main objective of action research in software engineering is to simultaneously solve a real-world problem and explore the experiences and results of problem-solving [68]. The participatory aspect of action research method allows the researcher to systematically determine, define the problem, and make a solution proposal in the context of an investigation. Moreover, it allows the researcher to actively participate in further steps of applying the solution in real-time which is termed as action [66] [67].

Action research gives the opportunity to work collaboratively with problem owners(concerned actors) at the organization and the possibility to propose, implement, and evaluate the solution in real-time. Problem owners are an inevitable part of action research since they share their skills, domain knowledge, and experiences [66] [67]. Besides, they can also share reflections or thoughts on a proposed solution. Practitioners can also explain why a proposed solution is not practically suitable for a particular context from their experience and domain knowledge. Action Research approach typically means that researchers engage with a company over time and during a process. In our study, a data scientist, a software developer, and a data analyst actively collaborated with the researchers to identify the problems, action strategies and engage continuously in problem-solving efforts. This study was developed over eight months in Company A and three months in Company B. In Company A, the researchers participated in weekly meetings with the teams for exploring the data pipelines and challenges involved in maintaining data pipelines. The researchers presented the findings of the study in an internal

Table 8.1: Activities in Action Research

Activity	A	B	Duration	Frequency
Data Pipeline Exploration	✓	✓	1 hour	Weekly meetings
Discussion with Data Scientists, Data Engineers and Data Analysts	✓	✓	1 hour	Weekly meetings
Data Pipeline Workshop	✓	✓	2 hours	Two times in Company A Two times in Company B
Change in design of data pipelines	✓	✓	NA	NA
Presentation of Findings	✓	✓	1 hour	Three times in Company A Two time in Company B
Check-in with Data pipeline in-charge	✓	✓	30 minutes	Bi-weekly in Company A Weekly in Company B
Preparation for Data Pipeline Implementation	✓	X	NA	NA
Implementation of changes	✓	X	6 hours	NA
Testing the implementation	✓	X	4 hours	NA

workshop in which the participants were Analytics Managers, Product Owners, Software Developers, Data Engineers and Data Scientists. Feedback from the participants was collected and recorded for further improvement of the study. Based on the findings, researchers implemented the solution in a small slice of one of the studied data pipelines. The software developer also guided the researchers during the implementation and testing of solutions. In Company B, the researchers participated in weekly meetings, in-depth discussions with practitioners including data analysts, data scientists and data engineers. Further, the study results were proposed in two steering committee meetings that approved the changes in design of data pipelines and waiting for the approval to start implementation. Table 8.1 summarizes the activities in which the researchers were involved in both the companies. Through action research, we have focused on exploring, understanding, and improving data pipelines in real contexts. The action research process cycle consists of five stages namely (1) diagnosis, (2) action planning and designing, (3) action taking, (4) evaluation, and (5) specifying learning [66] [67]. Each step of the research process cycle is described in detail below.

Problem diagnosis and field organization

The first step of action research, diagnosis focuses on identifying, understanding, and describing the problem from the studied organizational context [66]. We have two organizations that actively participated in the study. Therefore, the studied contexts involve two analytics R&D organizations with data scientists, software developers, ETL developers, and data analysts at large

software-intensive organizations within the telecommunication industry and automobile industry respectively. The researchers conducted a literature review analysis to identify and inform the challenges while maintaining data pipelines without monitoring and fault detection components.

Through consultation with problem owners, it was determined that there existed no automated monitoring in the data pipeline but rather fault detection was manually done by either data pipeline owner/flow-guardian in Company A and it was not at all monitored in the other. Typically, in Company B the errors in the data were accumulating over the steps in the data pipeline without being detected and the superuser of the data pipeline was detecting the data errors while receiving the data quality reports. According to the superuser who is also a data scientist, data quality is considerably low due to the problems in the data pipeline. The most emphasized problems include complete data file loss during transmission, data loss during data ingestion, and data errors generated through human involvement. One proposed action was to come up with a fault detection component and a mitigation strategy component in each of the links in the data pipeline that will identify potential issues in the data pipeline stages and suggest mitigation strategies to reduce the impact of fault thereby ensuring continuous and smooth flow of data.

Action planning and design

For solving the problems in the data pipeline, researchers proposed a solution, discussed in a workshop at Company A, and presented the findings before the steering committee for approval [66]. An initial step for incorporating automated fault detection practice and mitigation strategies involved understanding the steps in the existing data pipelines in the case companies. For the modeling of fault-tolerant data pipelines, this step is useful to inform about the actual and desired characteristics of data pipelines. At the same time, we did a literature review on fault-tolerant data pipelines implemented at large-scale industries like Google [142], Microsoft [15], Facebook [143] and LinkedIn [144].

Pipeline A1: Hardware Fault Detection using Machine Learning

Customers return the products from Company A to a screening center when they detect issues. Therefore, a machine learning model is deployed at the screening center to verify the fault in the products. Pipeline A1 which is spread across three zones namely A, B and C accomplish the hardware fault detection. Zone A is the data generation zone where the devices are located. Whenever a hardware problem is encountered, it is sent to Zone B where the machine learning component verifies whether the hardware component should be accepted by the company for repair or replacement. The machine learning component for hardware fault detection is developed at Zone C where company A is located. When the hardware component is returned to the screening center, i.e Zone B, data is logged in a raw data storage which is a centralized data lake for returned products. The data required for training the machine learning model is fetched from this raw data storage by sending the query and stores in a local data storage at Zone C for further processing. The data in the local data storage will be in the form of logs and they are different for different types of hardware products. Therefore, Zone C parses the logs and extracts the data required to train and test the machine learning models for hardware fault detection. There can be ill-formatted data, missing data, and non-numerical data which is mitigated using data imputation and categorical encoding which is collectively called data cleaning. The data thus prepared is validated once again before feeding it to the machine learning model. As we are only concerned about the data related activities here, all the model-oriented activities are not included in the pipeline A1. The model once deployed in the screening center is continuously monitored for evaluating the performance which is done in a semi-automatic fashion. i.e the key performance indicators(KPI) are automatically generated. However, the analysis and decision for retraining the model are taken after manual analysis of the KPIs and discussion with the subject matter experts(SME).

Pipeline A2: Network Analytics using Machine Learning

The data is collected from the centralized data collection system developed by the company as the data needs to be collected from the live network. Fig. 8.2 illustrates the data pipeline for network analytics. Although network an-

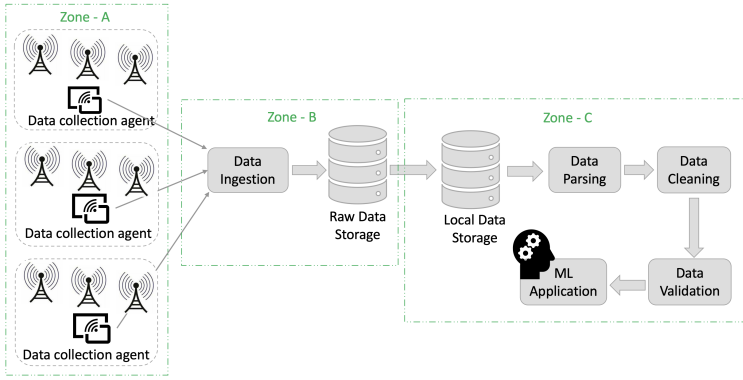

Figure 8.1: Pipeline A1 - Hardware Fault Detection using Machine Learning

alytics is a human activity, with pipeline A2, Company A is maximizing the automation for performing network analytics. Information preparation, identification of patterns, and anomalies are accomplished using the pipeline A2. This pipeline is spread across three zones A, B and C. Zone A is the data generation region where data sources, i.e base stations are located. Unlike other data pipelines, the data is collected from live customer networks. Therefore, special data collection agents are deployed in the customer networks to filter the customer sensitive data and collect the data required for performing network analytics. This data collection agent authorizes and authenticates itself through an access service mechanism. It sends an authentication request which is approved by the access service if the credentials are matching and then data access is given for that data collection agent. Zone B is collecting data and stores it in a raw data storage which is a central data repository. Data for performing analytics are fetched from this data repository through queries and stores it in a temporary local data storage at Zone C. Unlike other pipelines, Zone B and Zone C are parts of Company A. Data in the form of log files are then parsed to prepare data set for training and testing machine learning model. Data Validation checks are performed before feeding the machine learning model. If the checks fail, then data is again fetched from the local storage and the same activities in the pipeline are repeated and finally, it goes as input to the machine learning model which does network analytics.

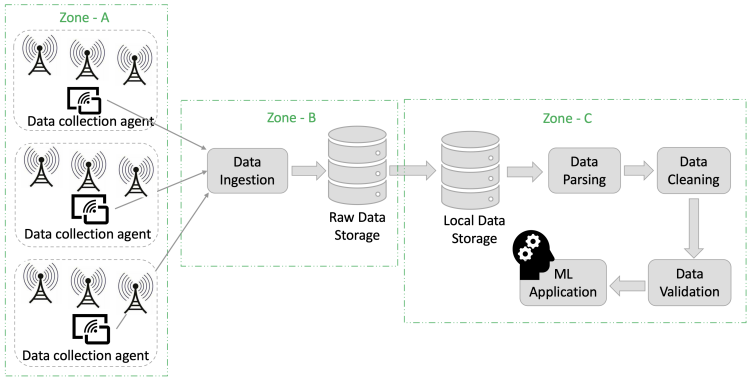

Figure 8.2: Pipeline A2 - Network Analytics using Machine Learning

Pipeline A3: CI/CD Data Integration Pipeline

The data pipeline illustrated in Fig. 8.3 is spread across three zones A, B, and C. Zone A and Zone C are outside Company A and Zone B is the company itself. This data pipeline A3 is the simplest pipeline in this study that serves the teams in company A who are working with data whenever they need it (With the term 'data', we mean the link from which the original data can be downloaded). The data pipeline components in Zone B collects data from multiple sources in Zone A and stores it in a data warehouse. Zone B collects two types of data dumps: internal(CI) and external(CD). The internal data dump is the data that is ingested by the teams inside the company A and external data dump is the data collected directly from the devices in the fields. The data ingestion method varies according to the data source. For instance, the data ingestion method for internal data and external data are different. The ingested data from multiple assorted sources are stored in the raw data storage for further use. The data can be in encrypted form which needs decryption before storing it in the refined data storage. The data archiver module sends encrypted data dump to the third-party services for decryption. Decoded links from the third party are transferred to data storage. Therefore, data from distributed sources are made available in a central location. For instance, if a team wants to access the external data from a continuous deployment zone for evaluating the key performance indicators or

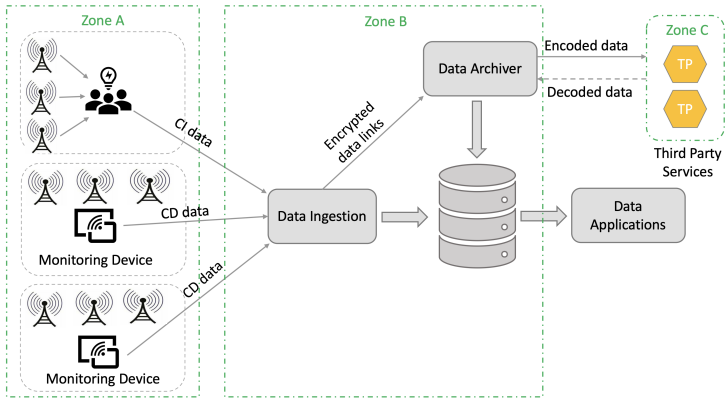

Figure 8.3: Pipeline A3 - CI/CD Data Integration Pipeline

internal data. The monitoring mechanism in the pipeline is manually carried out by the 'flow guardian' who is responsible for fixing the issues in the pipeline.

Pipeline B1: Data Quality Monitoring Pipeline

Data quality monitoring pipeline, B1 is spread across three zones, namely Zone A, Zone B, and Zone C. Data generation takes place at Zone A and it consists of all the manufacturing units, delivery, and repair centers distributed at different parts of the world. Zone B is a company which collects, ingest, aggregate, and store the data in a big data warehouse, marked as Refined Data Storage in Fig. 8.4. Zone C collects a part of data from the Refined data storage and creates data quality reports for the data scientists and data analysts. Company B where the study was conducted is at Zone C. Therefore, company B cannot exert much control over Zone B and Zone A.

The company at Zone B collects and stores four types of data and distributes it for teams as well as co-working organizations. Plant data, delivery data, warranty data, and repair data are the different types of data that are collected from sources such as manufacturing plants, service centers, delivery centers, and warranty offices. The manufacturing plants at Zone A will generate data for each product built there. The manufacturing data is collected from distributed manufacturing plants every 24 hours. However, not all the

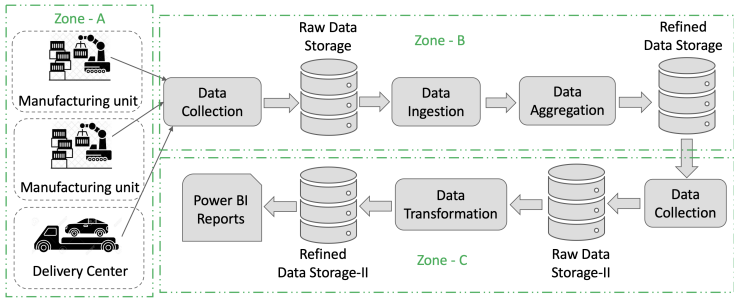

Figure 8.4: Pipeline B1 - Data Quality Monitoring Pipeline

data generated by the plants are collected by the data collection agent at Zone B. The Group Quality IT platform of the co-working organizations demand the data that needs to be collected from the plants. The data requested by the delivery centers are also collected and stored in the Refined Data Storage at Zone B. The data collected from different sources are in different formats and volumes. Therefore, data transfer mechanisms as well as data storage is different for all data sources. The data is ingested from the raw data storage and then transformed into a uniform format and stored in a data warehouse marked as Refined Data Storage which then acts as a supplier for teams as well as other organizations that demands data. For instance, the delivery centers need data about the products that are manufactured in the plants.

Company B at Zone C receives data collected and stored at Zone B and creates data quality reports which are used by the data scientists team for analyzing the product quality. For instance, the reports can be used to understand the model of the product that is sent to repair centers frequently. When the data quality is not satisfactory, the investigation is initiated and actions are taken to fix the data quality issues. Company B at Zone C access data at Zone B through a private network and then store it in another data storage marked as Refined Data Storage - II from where data scientists can access it for creating reports and training machine learning models.

All these data pipelines have different steps based on the use case. Therefore, we have created a conceptual model of data pipeline with some common steps as shown in fig. 8.5 for the easy presentation of analysis results. Based

on the data pipelines A1, A2, A3, and B1 we have collected data about the typical data pipeline faults and have classified the faults according to the steps where they appear. Some faults like change in data format repeatedly occur at various stages of the data pipeline. Table 8.2 illustrates the steps in a data pipeline, faults at each step, and the possible mitigation strategies. The faults marked in the table are the most typical ones causing data pipeline leakage and failure.

Action taking

The chosen method for the solution is implemented in the action taking step [66]. In this study, among the activities in action taking involved identification of fault detection and mitigation strategies in the data pipeline and the implementation of the same in a small piece of the existing data pipeline. Mostly a data scientist, a software developer, and the researchers conducted a weekly one-hour meeting at the organization's premises.

Here, the solution strategy we adopt is to include a fault detection component in the data pipeline marked as 'F' in fig. 8.5 which can detect the faults at a particular step and initiate mitigation action if defined. Sending alarm is a default mitigation strategy adopted by the data pipeline. Mitigation action cannot fix the issue, it can only ameliorate the impact of the fault. Some mitigation actions provide a temporary fix and for a permanent fix, human intervention might be needed. Therefore, sending alarms is done regardless of other mitigation strategies defined. Table 8.2 shows the data pipelines and their respective steps which initiate sending alarms immediately when a fault is encountered. Faults at different steps of the data pipeline vary according to the use case, dependency on other organizations, level of control on different zones, etc.

Since the incorporation of fault detection component at all stages of all four data pipelines is not an easy task to accomplish, we chose to select a small slice of the data pipeline for implementing fault detection and mitigation components as a pivot sample. Pipelines A1, A2, and B1 are spread across three zones and the case companies are in Zone C, it is difficult to get permission from authorities to incorporate new components. Therefore, we chose pipeline A3 which is the simplest pipeline in the study. Again, the most fault occurring slice of it was chosen for implementing our solution strategy. Thus, the link between Data Archiver and third party services was selected. Third-party

services decode the encrypted links sent to them and send back the decrypted links. However, due to some reasons decoded links are missed in the connector between data archiver and third parties resulting in the degradation of data quality.

For data pipeline B1, the implementation of fault detection components at the links in Zone B is pending approval, and Zone C is progressing. As Zone B and Zone C are completely different companies, many of the alarms received from manufacturing units and delivery centers were not shared with Company B. Therefore, sending alarms to both zones B and C was set as an action item. As the fault detection component is not implemented, alarms are generated only for a subset of faults like data sending job failure, change in data format, data generation software failure, and inactive data source. Nevertheless, these are some of the faults that occur at the very early steps of the data pipeline. Therefore, sending alarms for these itself is advantageous as the practitioners can easily identify the reason for unexpected changes in data products.

Evaluation

At the evaluation step, the effects of action taking are captured using methods such as focus groups, meetings, interviews, observations, etc [66]. In this study, while instances of evaluations also occurred throughout the design and implementation of fault detection components and mitigation strategies, explicit evaluations were captured during the initial development steps of the data pipeline model and its application on the organization's data pipeline. At this step, a summary of the feedback based on the implementation, open possibilities, and limitations was created and discussed between the researchers and the problem owners. Further, the software developer helped in identifying the repository where the fault detection and mitigation strategy should be coded without hampering the existing data flow through the pipeline and gave necessary instructions to do HBase coding with Python. During the session, the researcher collected data through observations and by asking a set of predefined questions to the data scientist and software developer. It is important to notice the evaluations were not only about the fault detection component but also the entire process of incorporating automated fault detection and mitigation strategies in an operational data pipeline.

Specific Learning

Lastly in action research, the general lessons are specified based on the evaluations to help decide on how to proceed e.g., a subsequent action research cycle [66]. Several improvements for the other parts of data pipelines were identified and agreed upon for further development. However, as the main finding, this study particularly reflects on the lessons gained from the process of incorporating fault detection components and mitigation strategies. The lessons learned are based on discussion and notes gathered during evaluations described above in Evaluation. Specific aspects .reflected upon and discussed as findings of the study include:

- Identification of faults and corresponding mitigation strategies at each step of the data pipeline
- Incorporation of automatic fault detection components and mitigation strategies in the data pipeline.
- Benefits and limitations of fault detection component and mitigation strategies in the data pipeline.

8.3 Findings

In this section, we present the findings from each of the two case companies involved in our study. In particular, we present different stages of a conceptual model of the data pipeline that we developed and identified the faults that occur at each step of the data pipeline and the corresponding mitigation strategies the companies can adopt to ameliorate the impact of the fault. The failures that we identify were derived from the empirical data and represent the typical failures that practitioners in the case companies experience in management of data.

Data Generation

Data generation is the first step of all data pipelines included in the study. Any fault at this step will be propagated to the successive steps of the data pipeline. Therefore, it is necessary to detect the faults and take corresponding mitigation actions to facilitate a smooth flow of data through the data pipelines. The possible faults at this step are:

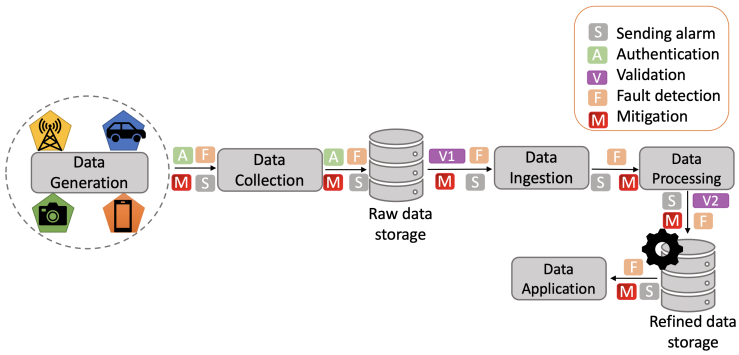

Figure 8.5: Conceptual model of data pipeline

Data Source Failure

The source generating data can fail due to many reasons. For instance, if a machine learning component is kept inside a vehicle, it will not get data when the vehicle crashes. In addition, the data generation component can fail.

Inactive data source

Data generation stops when the host device is inactive. This does not apply to all data pipelines. However, some data generation components are programmed to produce data only when in action. Thus, inactive data sources will not produce data. For instance, during the COVID-19 pandemic, manufacturing plants of pipeline B1 was inactive and didn't produce any data. In this case, the data source is not failed, but inactive.

Sending alarm is the only possible general mitigation strategy for all these faults in this step. Sending notification to reactivate the source is another mitigation strategy. However, all these needs human intervention.

Data Collection

Data from the data sources are often collected through a data collection agent or a data collection method that takes care of authentication and authorization services. Moreover, all the legal agreements and contracts are prepared and verified at this step.

Authentication failure

Authentication is the process of validating the identity of the agents trying to get access to the data generated by the data sources. Authentication failure restricts access to data and data pipeline breaks at the data collection step. Expired credentials are the main reason for the authentication failure. Therefore, as a mitigation strategy, functional user credentials which never expires can be obtained from the authorities. Otherwise, action should be taken to renew the credentials. Lack of legal agreements or expired contracts also leads to data accessibility issues. Setting up legal contracts is the action that needs to be taken. As this cannot be done automatically, sending an alarm to the concerned person is the only possible action.

Data sending job failure

Data is sent to the data collection device either continuously or as batches. All the data pipelines detailed in the previous section are processed in batches. A batch job is a scheduled program that is assigned to run without further user interaction. Failure of these jobs will also result in data pipeline breakage. In most of the batch processing systems, notification of failure is built in which makes the task of sending alarms trivial.

Unexpected data

Unexpected data is a very typical fault experienced by most of the practitioners. Partial failure of the above pipeline steps results in this fault. Moreover, improper communication between teams and the zones in the pipelines are other reasons for unexpected data. For instance, data pipeline B1 experiences this fault when a new source is added in Zone A without notifying the concerned data pipeline owner in Zone B.

Data Ingestion

Data ingestion is the process of collecting data from multiple assorted sources distributed around the globe. Failures in this stage can cause the disappearance of data from a specific data source which causes considerable pipeline performance degradation.

Incompatible ingestion methods

When a new source is added, it might generate data different from the existing sources leading. Thus, data pipelines fail to ingest the data from the new source. For instance, data pipeline B1 was recently added a new data source. However, the existing data ingestion methods fail to ingest the data from that source and as a result, the team is now building a new ingestion module specifically for that new data source.

Data extraction faults

Data is not always in the required format. Therefore, data extraction is performed to select and fetch the values required from available resources for further processing. Frequently, companies extract data to process it further, migrate the data to a data repository such as a data warehouse or a data lake or to further analyze it. This data extraction can be physical or logical. Data extraction methods fail to scrape the data properly when the type of resource is different than expected. Further, the data extraction method may fail to determine the relevancy of data and can scrape irrelevant data. Common mitigation strategies adopted include standardizing the data formats, converting the data to an acceptable format, defining a new data extraction method that can scrape relevant data from all different data formats which are considered as an ideal strategy.

Change in data formats

Change in data format is a common issue reported by all the four data pipelines. However, the extent of impact caused by this fault is different for different pipelines. For example, the impact is more on pipeline B1. Because Zone B and Zone C are different companies. The company at Zone C has a pipeline that is built to process certain types of data formats. When the data received from Zone B is in a completely different format, the data storage crashes causing data pipeline breakage and the further steps starve without getting data. This fault is a problem for all the successive data pipeline steps as well. Sending alarms while changing the format is the one possible mitigation strategy. Further, many companies use a versioning mechanism for data formats. These alarms are helpful for practitioners to adjust the data pipelines to receive the data in a new format.

Data Storage

Data Storage is a step that is often repeated in the data pipeline. Generally, there are two types of storage found in the aforementioned data pipelines: raw data storage and refined data storage based on the type of data stored in it. Based on the data access, storage is again classified as a central data repository and a local data repository.

Insufficient data storage

Insufficient data storage is a very common data pipeline fault especially when the data volume is very large. With the advent of Big data storage technologies, this fault can be fixed without much trouble. However, the most common mitigation strategy adopted by the case companies is to send an alarm to the developer and support team. They manage the storage space by deleting the temporary files, removing duplicates, etc. In pipelines A1 and A2, storage space is shared between teams. Therefore, storage capacity decreases when the data volume increases. An alarm is sent to all teams regarding the storage space and usually, it is managed manually by deleting the temporary files and data. All teams have read and write access. One possibility of fault is a team manipulating the original data and updating it with the result of their work leading to the loss of original data. The mitigation strategy implemented here is to allocate separate directories for each team using the shared cluster.

Data duplication

Storage of redundant data is another cause for the wastage of storage space. The main reason for redundant storage is different teams storing their copies of the same data in different directories of the same storage. Data de-duplication methods with Hive, Map-reduce, HDFS, etc are one of the best mitigation strategies that can be adopted.

Infrastructure failure

RAM Crash, Power-down, Hard-disk failure, etc are classified under infrastructure faults and are very critical. Sending alarms and waiting for the fix is the issue is the only possible mitigation strategy that can be adopted. To prevent data loss, data storage should be set as non-volatile. Recovering data

failed hard disk needs a skilled data recovery service.

Data Processing

Data processing is the collection and manipulation of data for producing meaningful insights. Data Processing is an umbrella term that can be used to represent several activities like data transformation, data cleaning, data preparation, data interpretation, etc.

Transformation faults

Data transformation is the process of converting data from one format or structure into another format or structure. When the data is converted from one form to another, some parts of the data are lost. Instead of defining a mitigation strategy, it is preferable to use a lossless data transformation technique whenever possible so that data loss can be prevented.

Unclear definitions and wrong interpretations

These are the least common fault encountered at the data processing step. For instance, a very common mistake in the machine learning model is the non-linear data misinterpreted as linear data and this leads to wrong model selection and eventually produces poor prediction results. Moreover, some values in the data are misinterpreted and taken for analysis without understanding the context. The possible mitigation strategy is to contact subject matter experts for correct interpretations and clarifying the doubts regarding the range of the data and possible values for a parameter.

Human errors

Human errors are very common and data pipelines maximize automation to minimize human errors. However, there are certain activities in the data pipeline that cannot be automated. For instance, the data labeling step in a machine learning pipeline cannot be completely automated. Therefore, the complete elimination of human errors is nearly impossible. Data validation is a mitigation strategy to reduce the impact of human errors. The data validation framework performs various checks to detect the errors in the data that are introduced by various sources.

Schema errors

Schema standardizes the data content. Schema errors occur where there is a problem with the structure or order of the file. Schema errors prevent the validation from being run in full because the file cannot be read. This means that errors cannot be traced to a particular record. Defining common schema and common language is another mitigation strategy that can be adopted. Schema validation is also a mitigation strategy adopted to identify schema errors.

Data Sink

Data sink is the final destination of data where the data pipeline ends. Data storage such as data lake, data warehouse, file systems or databases, data applications such as data visualization applications, data analysis models, machine learning models, etc are different types of data sinks in the data pipelines.

Dirty data

Data missing happens mostly during data transmission from one node to another. Data logs that are not returned in expected formats or a non-readable format are usually skipped causing missing data. For instance, data generation in pipeline A1 is carried out by the software component built in the hardware part sold to the customers. Due to the incompatible or obsolete data parsers in the data pipeline, data generated by these software components become unreadable and thus become unavailable for further processing. The mitigation strategy is to contact the team and they parse the logs after defining new parser or updating existing parser. There is a specific range of values for each of the parameters in the data. When the data is out of range, it results in severe data quality issues. Check the range of values against the parameters after discussion with subject matter experts. These errors can be detected through data validation. A common mitigation action is to replace with suitable statistical measure depending on the parameter.

Table 8.2: Common Faults and Mitigation Strategies

Data Pipeline Stage	Faults	A1	A2	A3	B1	Mitigation Strategies	Sending Alarms			
							A1	A2	A3	B1
Data Generation	Data source failure	X	X	X	X	Set a proxy which never fails		✓	✓	✓
	Inactive data source	X	X		X	Send notification to restart the source		✓	✓	✓
Data Collection	Authentication failure	X	X	X	X	Functional user credentials	✓	✓	✓	✓
	Data sending job failure	X	X	X	X	Send notification about failure	✓	✓	✓	✓
	Unexpected data	X	X		X	Send email to flow guardian	✓	✓		✓
Data Ingestion	Incompatible ingestion methods				X	Log the error, Define dedicated ingest modules		✓		✓
	Data Extraction faults	X	X	X	X	Conversion to acceptable format, Formalize the data, Define data extraction method that works for all data formats	✓	✓	✓	✓
	Change in data formats	X	X		X	Versioning mechanism	✓	✓	✓	✓
Data Storage	Insufficient storage	X	X	X	X	Alarm to the developer and then to support team	✓	✓	✓	✓
	Data duplication	X	X	X	X	Use of HDFS		✓		✓
	Infrastructure failure	X	X	X	X	Sending alarm to IT support	✓	✓	✓	✓
Data Processing	Transformation faults	X	X		X	Define lossless approaches				
	Unclear definitions and wrong interpretations	X	X	X	X	Contact SMEs				✓
	Human errors	X	X	X	X	Data validation				
	Schema errors	X	X		X	Define common schema and common language Write different parsers	✓	✓	✓	✓
Data Sink	Dirty data	X	X	X	X	Statistical methods, Data imputation techniques	✓	✓	✓	✓

8.4 Automated Data Pipeline Recovery

In this section, we detail the solution strategies that can be adopted for automated data pipeline recovery. The strategy is to include two connector level components in the data pipeline namely fault detection and mitigation. For the physical realization of this solution approach, we chose the data pipeline A3. Incorporating new components in all links is certainly not achievable within a short period due to practical difficulties. Therefore, we have chosen a small slice of the data pipeline as shown in fig. 8.6 which consists of two nodes and a connector in between. Data archiver and third party services are the two nodes and there is a connector in between them. The data archiver gets the encrypted data from the data ingestion module which is then transferred to the third parties for decryption.

Due to reasons such as inadequate bandwidth or insufficient resources some links are not sent back after decryption. Consequently, these links are never stored in the refined data storage and therefore not sent to the data application. Missing of links over time increases to an extent that it causes con-

siderable degradation in data quality. Therefore, we decided to keep a fault detection component in the connector between the two nodes.

Fault detection: Each data link is associated with a unique DUMP ID. When a new encrypted link is obtained from the data ingestion node, HBase table RBS DUMP FILE is scanned for the corresponding decrypted link. The encrypted link is sent to the third party if already decrypted link is not available in the HBase table. When the decryption is successful, third party services at Zone C sent the decrypted link back to Zone B. While obtaining the decrypted link, it is written back to the Hbase table RBS DUMP FILE by setting the response code as '200' and SERVICE FAILURE flag as 'NO' which is otherwise set as 'YES'. All those links stored in HBase before 24 hours and having a status of '200' represents failed data links. These links can be sent to third parties again on the next day for decryption together with the new links.

Mitigation Strategy: Access the HBase table RBS DUMP FILE and check for the data links stored before 24 hours and are having a response code of '200'. Fetch the DUMP IDs of all such data links and form small batches. Send them to the third parties using Kafka message service for decryption. Make a Kafka message that contains DUMP IDs back to third parties for reprocessing.

When the mitigation strategy was implemented for the first time, the number of failed dump ids was counted and it came around 32453 over 30 days. Before. implementing the fault detection and mitigation strategy, these dumps were skipped and were not taken for further processing. Consequently, the quality of the reports that are produced out of this data was poor. After implementing the fault detection and mitigation strategy, failed dumps are automatically resent as small batches along with the new dumps to the third parties for decryption.

8.5 Related Work

This section presents the most related previous studies on data pipelines conducted in both academia and large-scale industries including Facebook, LinkedIn, Google, and Microsoft.

The study in [144] by K. Goodhope et. al describes the building of a real-time activity data pipeline at LinkedIn. The study also highlights the design

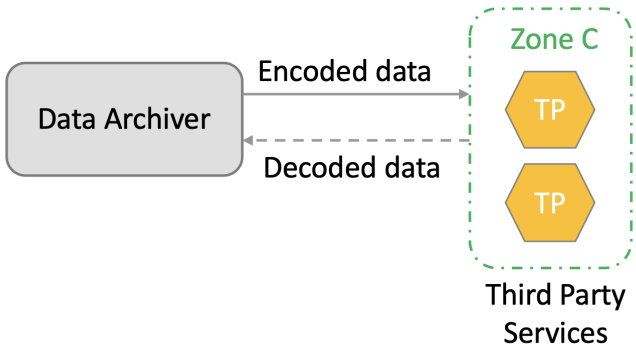

Figure 8.6: A small slice of data pipeline A3

and engineering challenges faced while moving the data pipeline of LinkedIn from a batch-oriented file aggregation mechanism to a publish-subscribe system in real-time. However, the paper has no explicit mention of whether the data pipeline is capable of recovering from failures.

A multiple case study conducted by Amershi, et. al at Microsoft [15] discusses a pipeline to address issues fundamental to the large-scale development and deployment of ML-based applications. The pipeline followed by the ML experts at Microsoft involves nine steps starting from data collection to model monitoring and the feedback loops. The pipeline stages include both data-oriented stages as well as model-oriented stages. However, the paper has no mention of monitoring and fault detection.

Tevfik et. al in [180] have proposed and validated an automated data pipeline that can manage multi-hop data transfers for moving a large volume of data among multiple sites. The data pipeline is resilient to failures and can recover automatically from a variety of networks, storage systems, software, and hardware failures. Besides, they have also shown a real-life data transfer involving thousands of large files and show that data pipeline works and is able to handle failures effectively.

A high-level component overview of a machine learning platform is discussed in [181]. In this paper, the authors state that creation, deployment, and maintenance of machine learning models require orchestration of components such as model generators, model validators, data validators, and infrastructure

for serving models in production. This study also recognizes the importance of fault detection and explains the impact of data errors.

A data pipeline for real-time data processing at Facebook is presented in [143]. The authors have identified five important design decisions that affect the ease of use, performance, fault tolerance, scalability, and correctness of data. In their paper, they discuss the usage of Scribe, a message bus that connects nodes in a data pipeline.

E. Caveness et. al discuss the TensorFlow data validation in continuous Machine Learning pipelines [142] at Google. The TensorFlow Data Validation(TFDV) architecture developed at Google is demonstrated in the paper which consists of a Data Analyzer component which computes statistics in a scalable fashion over large amounts of data, a Data Validator which finds anomalies in the data, and a Data Visualizer which provides visualizations of the statistics, schema, and the anomalies.

In summary, data pipelines have attracted significant interest from the software industry and research within a short time, and the notion of data pipelines among practitioners is growing. Besides, research related to data pipelines is emerging in the scientific literature. However, not many papers have discussed the data pipeline modeling, the essential components, faults at different stages of the data pipeline, and the definition of mitigation actions to reduce the impact of failure.

8.6 Conclusions

In this study, we explored four data pipelines in two companies and identified the faults at each step and the corresponding mitigation strategies. Based on a conceptual model of data pipeline we presented the concept of modeling a fault-tolerant data pipeline which can automatically detect common faults at various steps and take action to reduce the impact of the faults. In doing so, we can obtain a fault-tolerant, self-healing data pipeline. While the details in our study relate to each specific company, there are several implications that we think apply to more companies than those we studied. First, most of the faults identified are very typical and will apply to most of the data pipelines. Second, the classification of faults according to the stages helps companies to identify the faults that occur in their pipeline. Finally, the mitigation strategies we identified through our study help to reduce the impact of faults. For the first

time when implemented, the data pipeline will not be completely fault-tolerant as we cannot anticipate all the faults at various stages of the data pipeline. However, the list of faults and mitigation strategies can be updated when encountering new problems. While our study does not provide an exhaustive list of all the faults and mitigation strategies, we believe that the empirical insights presented here will be important for companies. In future works, we intend to further extend the list of. faults and mitigation strategies with more industrial cases of companies implementing data pipelines. Further, we intend to develop an end-to-end data pipeline with a fault detection component which does automated fault detection and mitigation.

CHAPTER 9

Concluding Remarks and Future Work

To sum up, this research investigates the data management practices and data pipeline model, especially for AI-enhanced embedded systems. The main goal of this study is to empirically identify the data management challenges encountered during the development and maintenance of AI-enhanced embedded systems, propose an improved data management approach and empirically validate the proposed approach. This thesis presents the motivation, procedures, and findings of our research conducted in the area of data management and data pipelines for AI-enhanced embedded systems. In this research, we have mainly focused on data management challenges and the adoption of robust data pipelines which is a mandatory component for practicing DataOps. We have identified the key data management challenges at each stage of the data pipeline through an exploratory case study. Through this study, we established the need for a better data management practice which lead us to the DataOps approach. As DataOps is an emerging practice, it is the least explored area considering the peer-reviewed literature and also not many companies were practicing it. Therefore, we conducted an extensive multi-vocal literature review including both peer-reviewed literature as well as grey literature to obtain an overview of the definition and its practices. We identified

the maturity stages through which the companies passed before the adoption of DataOps through multi-vocal literature review and multiple case studies at a company practicing DataOps. Not surprisingly we were able to observe the factors that were holding the companies from adopting DataOps. Further, we identified the critical and basic element required for practicing DataOps which is data pipelines. Despite the opportunities provided by data pipelines inevitably, there were several practical challenges such as organizational barriers, data quality challenges, and infrastructural problems that curbs companies from implementing and maintaining the data pipelines. Our study also concluded that there needs to be a domain-specific language for data pipelines for better communication between the teams within and across the companies. We then performed studies to understand the fundamental components required to build a robust data pipeline and designed a conceptual data pipeline model that enables minimum human intervention and maximum automation. We identified the typical faults at each stage of the data pipeline and corresponding mitigation strategies. Finally, we validated our data pipeline model by realizing a small slice of an existing operational data pipeline at one of the collaborating partner companies. We were able to increase the data quality by including more data dumps that were skipped before we implemented fault detection and mitigation strategies.

This thesis underlined the significance of data management practices such as DataOps and data pipelines for AI-enhanced embedded systems. The RQs framed in chapter 3 are addressed as follows.

RQ1: What are the challenges associated with data management in embedded system companies?

In order to answer this research question, we conducted the literature review and identified that no paper discussed the data management challenges for deep learning models. Therefore, through a multi-case study with case company B exploring use cases from 6 different domains, we identified 20 data management challenges encountered by practitioners while developing DL models and categorized them across the phases of the data pipeline. The challenges identified in this paper help practitioners foresee the roadblocks that may encounter while managing data for deep learning systems. Moreover, it provides an overview of data management challenges for deep learning models which was a gap in the literature. The study helps to identify the probable blind spots for the companies wishing to implement deep learning

models as well as guide future research.

RQ2: How can practices such as DataOps and Data Pipelines help address data management challenges?

To address this RQ, first, we conducted an exploratory case study at company B. From our investigation, we understood that the data management strategies were changed according to the evolving needs of the company. They have used ad-hoc data analysis, semi-automated data analysis, agile data science, continuous monitoring and testing, DataOps, etc for managing their data. The most recent data management approach they use for developing data products is DataOps. We also analyzed the key components required for adopting a better data management approach. Further, we identified the impediments that refrain the companies from climbing the steps towards DataOps. As DataOps is a relatively new concept, we did a multi-vocal literature review to derive a definition for DataOps.

Through our action research and exploratory case studies, we observed that there are challenges such as lack of robust data pipelines, data silos, overload on data flow guardians, and lack of orchestration in the existing operational data pipelines in the current data management practices. Further, we analyzed the challenges involved in developing and maintaining data pipelines and categorized them into three namely infrastructural, organizational, and data quality challenges. Through this study, we also identified that the data pipelines have a significant number of opportunities as well.

RQ3: What implications does AI have on data management and what practices can help address the development and maintenance of AI-enhanced embedded systems?

With the increased implementation of AI-enhanced embedded systems, companies started to encounter more problems associated with data management. It is fundamental to note that data is the backbone of AI models and errors in data can lead to significant performance degradation. Therefore, data management is essential to successfully deploy AI-enhanced embedded systems. Through action research and exploratory case study, we identified that the data pipeline is a good practice for data management. To mitigate the existing problems with the development and maintenance of data pipelines, we have proposed a conceptual model for data pipelines that maximizes automation and minimizes human intervention. To maximize automation, a fault detection component is employed at each stage of the data pipeline and the

fault detection component, in turn, calls corresponding mitigation strategy when faults are encountered. We have also identified the default mitigation strategy which is sending alarms.

We did action research at one of the companies to analyze the impact of improved data management practice. One of the data pipelines was selected for realizing the conceptual data pipeline model. Due to time constraints and company restrictions, we chose a small piece of the operational data pipeline and implemented the fault detection and mitigation strategies for that particular slice. We could observe that the implementation could save 32,453 data dumps which constituted 37% of the total data dumps flowing through that data pipeline link. This observation clearly illustrates that the improved data management practice using a conceptual model of data pipeline can increase the quality of data products. Moreover, the fault detection and mitigation strategies implemented at the pipeline stages maximize the automation thereby accelerating the development of AI-enhanced embedded systems.

9.1 Key contributions

The key contributions of this thesis are listed below.

Objective 1: To identify the data management challenges specific to Deep Learning models developed at embedded system companies

- Identification of the key challenges related to data management during the development of deep learning models.
- Categorization of data management challenges

Objective 2: To analyze how DataOps helps to address data management challenges

- Developed a definition for DataOps
- Identification of the key components for industrial data management
- Built an evolutionary model for data management practices
- Identification of the challenges at each stage of the evolution that are pulling companies backward from attaining the full potential.

Objective 3: To analyze how data pipelines help to address data management challenges in data applications and specifically AI-enhanced embedded systems

- Identification of the key elements for a conceptual data pipeline model
- Identification of the node level capabilities and connector level capabilities
- Identification the sequence of activities in the data pipeline model for data applications
- Identification the sequence of activities in the data pipeline model for AI-enhanced embedded systems
- Identification of opportunities with the implementation of data pipelines
- Identification of challenges during the implementation of data pipelines

Objective 4: To validate the data pipeline that maximizes automation and minimize human intervention

- Identification of the faults at each link of the data pipeline
- Identification of the mitigation strategies at each link of a data pipeline
- Identification of the default mitigation strategy
- Realize the data pipeline and analyze the impact

9.2 Future Work

In our research we have designed a conceptual data pipeline model, explored the opportunities and challenges encountered while building and maintaining them. We have also identified the typical faults at each stage of the data pipeline and the corresponding mitigation strategies. However, we adopted an inductive approach for validation due to a shortage of time. The validation is performed on a small piece of one of the data pipelines. To extend this further, we are planning to realize our conceptual data pipeline model so that companies can build their customized data pipelines by including the components to perform different activities that are demanded by their use

cases. Furthermore, we have plans to develop AI-based fault detection and mitigation system suitable for data pipelines.

References

[1] V. Mayer-Schönberger and K. Cukier, *Big data: A revolution that will transform how we live, work, and think*. Houghton Mifflin Harcourt, 2013.

[2] *Twitter seeing 6 billion api calls per day, 70k per second | techcrunch*, `https://techcrunch.com/2010/09/17/twitter-seeing-6-billion-api-calls-per-day-70k-per-second/?guccounter=1`, (Accessed on 04/13/2021).

[3] S. Sakr and S. Sakr, *Big Data 2.0 Processing Systems*. Springer, 2016.

[4] Å. Dragland, "Big data, for better or worse: 90% of world's data generated over last two years," *Science Daily*, vol. 22, 2013.

[5] *Designing data products. the 15 faces of data products are a... | by simon o'regan | towards data science*, `https://towardsdatascience.com/designing-data-products-b6b93edf3d23`, (Accessed on 03/28/2021).

[6] J. Bosch and H. H. Olsson, "Digital for real: A multicase study on the digital transformation of companies in the embedded systems domain," *Journal of Software: Evolution and Process*, e2333, 2021.

[7] *Big data management: How organizations create and implement data strategies*, `https://datascience.foundation/sciencewhitepaper/big-data-management-how-organizations-create-and-implement-data-strategies`, (Accessed on 03/27/2021).

[8] R. Abousleiman, G. Qu, and O. Rawashdeh, "North atlantic right whale contact call detection," *arXiv preprint arXiv:1304.7851*, 2013.

[9] C. Hill, R. Bellamy, T. Erickson, and M. Burnett, "Trials and tribulations of developers of intelligent systems: A field study," in *2016 IEEE Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, IEEE, 2016, pp. 162–170.

[10] R. S. S. Kumar, A. Wicker, and M. Swann, "Practical machine learning for cloud intrusion detection: Challenges and the way forward," in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 2017, pp. 81–90.

[11] T. Raeder, O. Stitelman, B. Dalessandro, C. Perlich, and F. Provost, "Design principles of massive, robust prediction systems," in *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2012, pp. 1357–1365.

[12] J. Schleier-Smith, "An architecture for agile machine learning in real-time applications," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2015, pp. 2059–2068.

[13] S. Tata, A. Popescul, M. Najork, M. Colagrosso, J. Gibbons, A. Green, A. Mah, M. Smith, D. Garg, C. Meyer, *et al.*, "Quick access: Building a smart experience for google drive," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1643–1651.

[14] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, M. Young, J.-F. Crespo, and D. Dennison, "Hidden technical debt in machine learning systems," in *Advances in neural information processing systems*, 2015, pp. 2503–2511.

[15] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, and T. Zimmermann, "Software engineering for machine learning: A case study," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, IEEE, 2019, pp. 291–300.

[16] A. Kaplan and M. Haenlein, "Siri, siri, in my hand: Who's the fairest in the land? on the interpretations, illustrations, and implications of artificial intelligence," *Business Horizons*, vol. 62, no. 1, pp. 15–25, 2019.

[17] V. Kepuska and G. Bohouta, "Next-generation of virtual personal assistants (microsoft cortana, apple siri, amazon alexa and google home)," in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, 2018, pp. 99–103.

[18] O. M. Parkhi, A. Vedaldi, and A. Zisserman, "Deep face recognition," 2015.

[19] F. Almeida and C. Calistru, "The main challenges and issues of big data management," *International Journal of Research Studies in Computing*, vol. 2, no. 1, pp. 11–20, 2013.

[20] B. A. Devlin and P. T. Murphy, "An architecture for a business and information system," *IBM systems Journal*, vol. 27, no. 1, pp. 60–80, 1988.

[21] J. Gray, "Evolution of data management," *Computer*, vol. 29, no. 10, pp. 38–46, 1996.

[22] L. Wu, G. Barash, and C. Bartolini, "A service-oriented architecture for business intelligence," in *IEEE international conference on service-oriented computing and applications (SOCA'07)*, IEEE, 2007, pp. 279–285.

[23] I. Suriarachchi and B. Plale, "Crossing analytics systems: A case for integrated provenance in data lakes," in *2016 IEEE 12th International Conference on e-Science (e-Science)*, IEEE, 2016, pp. 349–354.

[24] J. Bosch, H. H. Olsson, and I. Crnkovic, "Engineering ai systems: A research agenda," in *Artificial Intelligence Paradigms for Smart Cyber-Physical Systems*, IGI Global, 2021, pp. 1–19.

[25] L. E. Lwakatare, A. Raj, J. Bosch, H. H. Olsson, and I. Crnkovic, "A taxonomy of software engineering challenges for machine learning systems: An empirical investigation," in *International Conference on Agile Software Development*, Springer, Cham, 2019, pp. 227–243.

[26] A. Arpteg, B. Brinne, L. Crnkovic-Friis, and J. Bosch, "Software engineering challenges of deep learning," in *2018 44th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, IEEE, 2018, pp. 50–59.

[27] N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich, "Data lifecycle challenges in production machine learning: A survey," *ACM SIGMOD Record*, vol. 47, no. 2, pp. 17–28, 2018.

[28] ——, "Data management challenges in production machine learning," in *Proceedings of the 2017 ACM International Conference on Management of Data*, 2017, pp. 1723–1726.

[29] J. Bosch, H. H. Olsson, and I. Crnkovic, "It takes three to tango: Requirement, outcome/data, and ai driven development.," in *SiBW*, 2018, pp. 177–192.

[30] *How to create data products - from data scientist to business owner*, `https://www.tableau.com/learn/whitepapers/turn-data-products-data-scientist-data-business-owner`, (Accessed on 03/28/2021).

[31] A. A. Tole *et al.*, "Big data challenges," *Database systems journal*, vol. 4, no. 3, pp. 31–40, 2013.

[32] A. Labrinidis and H. V. Jagadish, "Challenges and opportunities with big data," *Proceedings of the VLDB Endowment*, vol. 5, no. 12, pp. 2032–2033, 2012.

[33] A. Oguntimilehin and E. Ademola, "A review of big data management, benefits and challenges," *A Review of Big Data Management, Benefits and Challenges*, vol. 5, no. 6, pp. 1–7, 2014.

[34] X. L. Dong and T. Rekatsinas, "Data integration and machine learning: A natural synergy," in *Proceedings of the 2018 international conference on management of data*, 2018, pp. 1645–1650.

[35] A. Halevy, F. Korn, N. F. Noy, C. Olston, N. Polyzotis, S. Roy, and S. E. Whang, "Goods: Organizing google's datasets," in *Proceedings of the 2016 International Conference on Management of Data*, 2016, pp. 795–806.

[36] J. Manyika, M. Chui, B. Brown, J. Bughin, R. Dobbs, C. Roxburgh, A. Hung Byers, *et al.*, *Big data: The next frontier for innovation, competition, and productivity*. McKinsey Global Institute, 2011.

[37] L. E. Lwakatare, T. Karvonen, T. Sauvola, P. Kuvaja, H. H. Olsson, J. Bosch, and M. Oivo, "Towards devops in the embedded systems domain: Why is it so hard?" In *2016 49th hawaii international conference on system sciences (hicss)*, IEEE, 2016, pp. 5437–5446.

[38] A. Siddiqa, I. A. T. Hashem, I. Yaqoob, M. Marjani, S. Shamshirband, A. Gani, and F. Nasaruddin, "A survey of big data management: Taxonomy and state-of-the-art," *Journal of Network and Computer Applications*, vol. 71, pp. 151–166, 2016.

[39] M. Xiaofeng and C. Xiang, "Big data management: Concepts, techniques and challenges," *Journal of computer research and development*, vol. 50, no. 1, p. 146, 2013.

[40] X.-W. Chen and X. Lin, "Big data deep learning: Challenges and perspectives," *IEEE access*, vol. 2, pp. 514–525, 2014.

[41] J. Al-Jaroodi, B. Hollein, and N. Mohamed, "Applying software engineering processes for big data analytics applications development," in *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*, IEEE, 2017, pp. 1–7.

[42] S. R. Dharmapal and K. T. Sikamani, "Big data analytics using agile model," in *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, IEEE, 2016, pp. 1088–1091.

[43] P. Simon, *Analytics: the agile way*. John Wiley & Sons, 2017.

[44] R. Alt-Simmons, *Agile by Design: An Implementation Guide to Analytic Lifecycle Management*. John Wiley & Sons, 2015.

[45] L. Bass, I. Weber, and L. Zhu, *DevOps: A software architect's perspective*. Addison-Wesley Professional, 2015.

[46] L. Zhu, L. Bass, and G. Champlin-Scharff, "Devops and its practices," *IEEE Software*, vol. 33, no. 3, pp. 32–34, 2016.

[47] *Dataops - wikipedia*, `https://en.wikipedia.org/wiki/DataOps`, (Accessed on 04/07/2021).

[48] J. Ereth, "Dataops-towards a definition.," in *LWDA*, 2018, pp. 104–112.

[49] M. Rodriguez, L. J. P. de Araújo, and M. Mazzara, "Good practices for the adoption of dataops in the software industry," in *Journal of Physics: Conference Series*, IOP Publishing, vol. 1694, 2020, p. 012 032.

[50] *Dataops - devops for big data and analytics | xenonstack*, https://www.xenonstack.com/insights/what-is-dataops/, (Accessed on 01/14/2020).

[51] *What is data pipeline: Components, types, and use cases | altexsoft*, https://www.altexsoft.com/blog/data-pipeline-components-and-types/, (Accessed on 03/08/2021).

[52] *What is data pipeline: Components, types, and use cases | altexsoft*, https://www.altexsoft.com/blog/data-pipeline-components-and-types/, (Accessed on 03/07/2021).

[53] *The importance and benefits of a data pipeline | xplenty*, https://www.xplenty.com/blog/what-is-a-data-pipeline/, (Accessed on 03/07/2021).

[54] *Why you shouldn't build your own data pipeline | blog | fivetran*, https://fivetran.com/blog/why-you-shouldnt-build-your-own-data-pipeline, (Accessed on 03/08/2021).

[55] S. B. Merriam *et al.*, "Introduction to qualitative research," *Qualitative research in practice: Examples for discussion and analysis*, vol. 1, no. 1, pp. 1–17, 2002.

[56] V. A. Lambert and C. E. Lambert, "Qualitative descriptive research: An acceptable design," *Pacific Rim International Journal of Nursing Research*, vol. 16, no. 4, pp. 255–256, 2012.

[57] R. Bogdan and S. K. Biklen, *Qualitative research for education*. Allyn & Bacon Boston, MA, 1997.

[58] C. B. Seaman, "Qualitative methods in empirical studies of software engineering," *IEEE Transactions on software engineering*, vol. 25, no. 4, pp. 557–572, 1999.

[59] *Collaborating partners – software center*, https://www.software-center.se/partners/, (Accessed on 04/14/2021).

[60] J. Sutton and Z. Austin, "Qualitative research: Data collection, analysis, and management," *The Canadian journal of hospital pharmacy*, vol. 68, no. 3, p. 226, 2015.

[61] J. Singer, S. E. Sim, and T. C. Lethbridge, "Software engineering data collection for field studies," in *Guide to Advanced Empirical Software Engineering*, Springer, 2008, pp. 9–34.

[62] K. F. Hyde, "Recognising deductive processes in qualitative research," *Qualitative market research: An international journal*, 2000.

[63] S. Crowe, K. Cresswell, A. Robertson, G. Huby, A. Avery, and A. Sheikh, "The case study approach," *BMC medical research methodology*, vol. 11, no. 1, pp. 1–9, 2011.

[64] D. E. Perry, S. E. Sim, and S. M. Easterbrook, "Case studies for software engineers," in *Proceedings. 26th International Conference on Software Engineering*, IEEE, 2004, pp. 736–738.

[65] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical software engineering*, vol. 14, no. 2, p. 131, 2009.

[66] K. Petersen, C. Gencel, N. Asghari, D. Baca, and S. Betz, "Action research as a model for industry-academia collaboration in the software engineering context," in *Proceedings of the 2014 international workshop on Long-term industrial collaboration on software engineering*, 2014, pp. 55–62.

[67] J. McKay and P. Marshall, "The dual imperatives of action research," *Information Technology & People*, 2001.

[68] S. Easterbrook, J. Singer, M.-A. Storey, and D. Damian, "Selecting empirical methods for software engineering research," in *Guide to advanced empirical software engineering*, Springer, 2008, pp. 285–311.

[69] D. Coghlan, "Action research: Exploring perspectives on a philosophy of practical knowing," *Academy of Management Annals*, vol. 5, no. 1, pp. 53–87, 2011.

[70] A. Strauss and J. Corbin, *Basics of qualitative research techniques*. Citeseer, 1998.

[71] S. Baskarada, "Qualitative case study guidelines," *Baškarada, S.(2014). Qualitative case studies guidelines. The Qualitative Report*, vol. 19, no. 40, pp. 1–25, 2014.

[72] C. Robson, *Real world research: A resource for social scientists and practitioner-researchers*. Wiley-Blackwell, 2002.

[73] K. Louise Barriball and A. While, "Collecting data using a semi-structured interview: A discussion paper," *Journal of advanced nursing*, vol. 19, no. 2, pp. 328–335, 1994.

[74] N. Cowie, "Observation," in *Qualitative research in applied linguistics*, Springer, 2009, pp. 165–181.

[75] S. Owen, P. Brereton, and D. Budgen, "Protocol analysis: A neglected practice," *Communications of the ACM*, vol. 49, no. 2, pp. 117–122, 2006.

[76] A. Von Mayrhauser and A. M. Vans, "Identification of dynamic comprehension processes during large scale maintenance," *IEEE Transactions on Software Engineering*, vol. 22, no. 6, pp. 424–437, 1996.

[77] A. Karahasanoviæ, B. Anda, E. Arisholm, S. E. Hove, M. Jørgensen, D. I. Sjøberg, and R. Welland, "Collecting feedback during software engineering experiments," *Empirical Software Engineering*, vol. 10, no. 2, pp. 113–147, 2005.

[78] P. Brereton, B. A. Kitchenham, D. Budgen, M. Turner, and M. Khalil, "Lessons from applying the systematic literature review process within the software engineering domain," *Journal of systems and software*, vol. 80, no. 4, pp. 571–583, 2007.

[79] V. Garousi, M. Felderer, and M. V. Mäntylä, "Guidelines for including grey literature and conducting multivocal literature reviews in software engineering," *Information and Software Technology*, vol. 106, pp. 101–121, 2019.

[80] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering–a systematic literature review," *Information and software technology*, vol. 51, no. 1, pp. 7–15, 2009.

[81] B. Kitchenham, R. Pretorius, D. Budgen, O. P. Brereton, M. Turner, M. Niazi, and S. Linkman, "Systematic literature reviews in software engineering–a tertiary study," *Information and software technology*, vol. 52, no. 8, pp. 792–805, 2010.

[82] L. Wong, "Data analysis in qualitative research: A brief guide to using nvivo," *Malaysian family physician: the official journal of the Academy of Family Physicians of Malaysia*, vol. 3, no. 1, p. 14, 2008.

[83] I. Dey, *Qualitative data analysis: A user friendly guide for social scientists*. Routledge, 2003.

[84] D. S. Cruzes and T. Dyba, "Recommended steps for thematic synthesis in software engineering," in *2011 international symposium on empirical software engineering and measurement*, IEEE, 2011, pp. 275–284.

[85] J. Saldaña, *The coding manual for qualitative researchers*. SAGE Publications Limited, 2021.

[86] L. Bickman and D. J. Rog, *The SAGE handbook of applied social research methods*. Sage publications, 2008.

[87] S. Zhang, L. Yao, A. Sun, and Y. Tay, "Deep learning based recommender system: A survey and new perspectives," *ACM Computing Surveys (CSUR)*, vol. 52, no. 1, pp. 1–38, 2019.

[88] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[89] C. Szegedy, A. Toshev, and D. Erhan, "Deep neural networks for object detection," in *Advances in neural information processing systems*, 2013, pp. 2553–2561.

[90] X. Liu, J. Gao, X. He, L. Deng, K. Duh, and Y.-Y. Wang, "Representation learning using multi-task deep neural networks for semantic classification and information retrieval," 2015.

[91] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal processing magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[92] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[93] R. Miotto, F. Wang, S. Wang, X. Jiang, and J. T. Dudley, "Deep learning for healthcare: Review, opportunities and challenges," *Briefings in bioinformatics*, vol. 19, no. 6, pp. 1236–1246, 2018.

[94] T. Wang, C.-K. Wen, H. Wang, F. Gao, T. Jiang, and S. Jin, "Deep learning for wireless physical layer: Opportunities and challenges," *China Communications*, vol. 14, no. 11, pp. 92–111, 2017.

[95] K. Pei, Y. Cao, J. Yang, and S. Jana, "Deepxplore: Automated white-box testing of deep learning systems," in *proceedings of the 26th Symposium on Operating Systems Principles*, 2017, pp. 1–18.

[96] M. Helmstaedter, K. L. Briggman, S. C. Turaga, V. Jain, H. S. Seung, and W. Denk, "Connectomic reconstruction of the inner plexiform layer in the mouse retina," *Nature*, vol. 500, no. 7461, pp. 168–174, 2013.

[97] H. Y. Xiong, B. Alipanahi, L. J. Lee, H. Bretschneider, D. Merico, R. K. Yuen, Y. Hua, S. Gueroussov, H. S. Najafabadi, T. R. Hughes, *et al.*, "The human splicing code reveals new insights into the genetic determinants of disease," *Science*, vol. 347, no. 6218, 2015.

[98] J. Ma, R. P. Sheridan, A. Liaw, G. E. Dahl, and V. Svetnik, "Deep neural nets as a method for quantitative structure–activity relationships," *Journal of chemical information and modeling*, vol. 55, no. 2, pp. 263–274, 2015.

[99] T. Ciodaro, D. Deva, J. De Seixas, and D. Damazio, "Online particle detection with neural networks based on topological calorimetry information," in *Journal of physics: conference series*, IOP Publishing, vol. 368, 2012, p. 012 030.

[100] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," *Communications of the ACM*, vol. 60, no. 6, pp. 84–90, 2017.

[101] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural computation*, vol. 1, no. 4, pp. 541–551, 1989.

[102] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[103] W. Wang, M. Zhang, G. Chen, H. Jagadish, B. C. Ooi, and K.-L. Tan, "Database meets deep learning: Challenges and opportunities," *ACM SIGMOD Record*, vol. 45, no. 2, pp. 17–22, 2016.

[104] *How 'deep learning' works at apple, beyond — the information*, `https://www.theinformation.com/articles/how-deep-learning-works-at-apple-beyond`, (Accessed on 11/21/2020).

[105] N. Jones, "Computer science: The learning machines," *Nature News*, vol. 505, no. 7482, p. 146, 2014.

[106] M. M. Najafabadi, F. Villanustre, T. M. Khoshgoftaar, N. Seliya, R. Wald, and E. Muharemagic, "Deep learning applications and challenges in big data analytics," *Journal of Big Data*, vol. 2, no. 1, p. 1, 2015.

[107] G. Tur and R. De Mori, *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011.

[108] P. Baxter, S. Jack, *et al.*, "Qualitative case study methodology: Study design and implementation for novice researchers," *The qualitative report*, vol. 13, no. 4, pp. 544–559, 2008.

[109] M. Maguire and B. Delahunt, "Doing a thematic analysis: A practical, step-by-step guide for learning and teaching scholars.," *AISHE-J: The All Ireland Journal of Teaching and Learning in Higher Education*, vol. 9, no. 3, 2017.

[110] H. Atwal, "The dataops factory," in *Practical DataOps*, Springer, 2020, pp. 249–266.

[111] L. E. Lwakatare, P. Kuvaja, and M. Oivo, "An exploratory study of devops extending the dimensions of devops with practices," *ICSEA 2016*, vol. 104, 2016.

[112] M. Shahin, M. A. Babar, and L. Zhu, "Continuous integration, delivery and deployment: A systematic review on approaches, tools, challenges and practices," *IEEE Access*, vol. 5, pp. 3909–3943, 2017.

[113] S. Ilieva, P. Ivanov, and E. Stefanova, "Analyses of an agile methodology implementation," in *Proceedings. 30th Euromicro Conference, 2004.*, IEEE, 2004, pp. 326–333.

[114] P. R. Sahoo and A. Premchand, "Dataops in manufacturing and utilities industries," 2019.

[115] J. Bosch, *Speed, data, and ecosystems: Excelling in a software-driven world*. CRC press, 2017.

[116] V. Garousi, M. Felderer, and M. V. Mäntylä, "The need for multivocal literature reviews in software engineering: Complementing systematic literature reviews with grey literature," in *Proceedings of the 20th international conference on evaluation and assessment in software engineering*, ACM, 2016, p. 26.

[117] R. T. Ogawa and B. Malen, "Towards rigor in reviews of multivocal literatures: Applying the exploratory case study method," *Review of educational research*, vol. 61, no. 3, pp. 265–286, 1991.

[118] *3 reasons why dataops is essential for big data success | ibm big data & analytics hub*, `https://www.ibmbigdatahub.com/blog/3-reasons-why-dataops-essential-big-data-success`, (Accessed on 01/24/2020).

[119] *From devops to dataops - dataops tools transformation | tamr*, `https://www.tamr.com/blog/from-devops-to-dataops-by-andy-palmer/`, (Accessed on 01/14/2020).

[120] *Data ops*, `https://www.gartner.com/en/information-technology/glossary/data-ops`, (Accessed on 01/14/2020).

[121] *Dataops and the dataops manifesto - odsc - open data science - medium*, `https://medium.com/@ODSC/dataops-and-the-dataops-manifesto-fc6169c02398`, (Accessed on 01/14/2020).

[122] *The dataops manifesto*, `https://www.dataopsmanifesto.org/`, (Accessed on 01/14/2020).

[123] *Dataops: Changing the world one organization at a time | zdnet*, `https://www.zdnet.com/article/dataops-changing-the-world-one-organization-at-a-time/`, (Accessed on 01/14/2020).

[124] *Dataops is not just devops for data - data-ops - medium*, `https://medium.com/data-ops/dataops-is-not-just-devops-for-data-6e03083157b7`, (Accessed on 12/20/2019).

[125] *Diving into dataops: The underbelly of modern data pipelines*, `https://www.eckerson.com/articles/diving-into-dataops-the-underbelly-of-modern-data-pipelines`, (Accessed on 01/14/2020).

[126] *Dataops in seven steps - data-ops - medium*, `https://medium.com/data-ops/dataops-in-7-steps-f72ff2b37812`, (Accessed on 01/25/2020).

[127] *Dataops: More than devops for data pipelines*, https://www.eckerson.com/articles/dataops-more-than-devops-for-data-pipelines, (Accessed on 01/25/2020).

[128] *What is dataops? everything you need to know | oracle data science*, https://blogs.oracle.com/datascience/what-is-dataops-everything-you-need-to-know, (Accessed on 01/25/2020).

[129] *What is dataops? - dataops zone*, https://dataopszone.com/what-is-dataops/, (Accessed on 01/25/2020).

[130] *Get ready for dataops - dataversity*, https://www.dataversity.net/get-ready-for-dataops/, (Accessed on 01/25/2020).

[131] H. Jagadish, J. Gehrke, A. Labrinidis, Y. Papakonstantinou, J. M. Patel, R. Ramakrishnan, and C. Shahabi, "Big data and its technical challenges," *Communications of the ACM*, vol. 57, no. 7, pp. 86–94, 2014.

[132] *The emergence of dataops empowers the future of data management | analytics insight*, https://www.analyticsinsight.net/emergence-dataops-empowers-future-data-management/, (Accessed on 01/24/2020).

[133] T. H. Davenport and J. Dyché, "Big data in big companies," *International Institute for Analytics*, vol. 3, 2013.

[134] B. Marr, *Big data in practice: how 45 successful companies used big data analytics to deliver extraordinary results*. John Wiley & Sons, 2016.

[135] L. Cai and Y. Zhu, "The challenges of data quality and data quality assessment in the big data era," *Data science journal*, vol. 14, 2015.

[136] C. Batini, A. Rula, M. Scannapieco, and G. Viscusi, "From data quality to big data quality," in *Big Data: Concepts, Methodologies, Tools, and Applications*, IGI Global, 2016, pp. 1934–1956.

[137] S. Kaisler, F. Armour, J. A. Espinosa, and W. Money, "Big data: Issues and challenges moving forward," in *2013 46th Hawaii International Conference on System Sciences*, IEEE, 2013, pp. 995–1004.

[138] K. Raman, A. Swaminathan, J. Gehrke, and T. Joachims, "Beyond myopic inference in big data pipelines," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2013, pp. 86–94.

[139] P. Jovanovic, S. Nadal, O. Romero, A. Abelló, and B. Bilalli, "Quarry: A user-centered big data integration platform," *Information Systems Frontiers*, pp. 1–25, 2020.

[140] N. Marz and J. Warren, *Big Data: Principles and best practices of scalable real-time data systems*. New York; Manning Publications Co., 2015.

[141] A. G. Carretero, F. Gualo, I. Caballero, and M. Piattini, "Mamd 2.0: Environment for data quality processes implantation based on iso 8000-6x and iso/iec 33000," *Computer Standards & Interfaces*, vol. 54, pp. 139–151, 2017.

[142] E. Caveness, P. S. GC, Z. Peng, N. Polyzotis, S. Roy, and M. Zinkevich, "Tensorflow data validation: Data analysis and validation in continuous ml pipelines," in *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, 2020, pp. 2793–2796.

[143] G. J. Chen, J. L. Wiener, S. Iyer, A. Jaiswal, R. Lei, N. Simha, W. Wang, K. Wilfong, T. Williamson, and S. Yilmaz, "Realtime data processing at facebook," in *Proceedings of the 2016 International Conference on Management of Data*, 2016, pp. 1087–1098.

[144] K. Goodhope, J. Koshy, J. Kreps, N. Narkhede, R. Park, J. Rao, and V. Y. Ye, "Building linkedin's real-time activity data pipeline.," *IEEE Data Eng. Bull.*, vol. 35, no. 2, pp. 33–45, 2012.

[145] A. Munappy, J. Bosch, H. H. Olsson, A. Arpteg, and B. Brinne, "Data management challenges for deep learning," in *2019 45th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, IEEE, 2019, pp. 140–147.

[146] M. Pathirage, *Kappa architecture - where every thing is a stream*, `http://milinda.pathirage.org/kappa-architecture.com/`, (Accessed on 09/28/2020).

[147] J. M. Verner, J. Sampson, V. Tosic, N. A. Bakar, and B. A. Kitchenham, "Guidelines for industrially-based multiple case studies in software engineering," in *2009 Third International Conference on Research Challenges in Information Science*, IEEE, 2009, pp. 313–324.

[148] P. Burnard, "A method of analysing interview transcripts in qualitative research," *Nurse education today*, vol. 11, no. 6, pp. 461–466, 1991.

[149] J. A. Maxwell, "Designing a qualitative study," *The SAGE handbook of applied social research methods*, vol. 2, pp. 214–253, 2008.

[150] P. O'Donovan, K. Leahy, K. Bruton, and D. T. O'Sullivan, "An industrial big data pipeline for data-driven analytics maintenance applications in large-scale smart manufacturing facilities," *Journal of Big Data*, vol. 2, no. 1, p. 25, 2015.

[151] C. P. Chen and C.-Y. Zhang, "Data-intensive applications, challenges, techniques and technologies: A survey on big data," *Information sciences*, vol. 275, pp. 314–347, 2014.

[152] T. C. Redman, "Data's credibility problem," *Harvard Business Review*, vol. 91, no. 12, pp. 84–88, 2013.

[153] B. Carlo, B. Daniele, C. Federico, and G. Simone, "A data quality methodology for heterogeneous data," *International Journal of Database Management Systems*, vol. 3, no. 1, 2011.

[154] M. W. Van Alstyne, G. G. Parker, and S. P. Choudary, "Pipelines, platforms, and the new rules of strategy," *Harvard business review*, vol. 94, no. 4, pp. 54–62, 2016.

[155] R. Matheus, M. Janssen, and D. Maheshwari, "Data science empowering the public: Data-driven dashboards for transparent and accountable decision-making in smart cities," *Government Information Quarterly*, p. 101 284, 2018.

[156] J. G. Stadler, K. Donlon, J. D. Siewert, T. Franken, and N. E. Lewis, "Improving the efficiency and ease of healthcare analysis through use of data visualization dashboards," *Big Data*, vol. 4, no. 2, pp. 129–135, 2016.

[157] G. Gautam and D. Yadav, "Sentiment analysis of twitter data using machine learning approaches and semantic analysis," in *2014 Seventh International Conference on Contemporary Computing (IC3)*, IEEE, 2014, pp. 437–442.

[158] S. M. S. Tanzil, W. Hoiles, and V. Krishnamurthy, "Adaptive scheme for caching youtube content in a cellular network: Machine learning approach," *IEEE Access*, vol. 5, pp. 5870–5881, 2017.

[159] P. Covington, J. Adams, and E. Sargin, "Deep neural networks for youtube recommendations," in *Proceedings of the 10th ACM conference on recommender systems*, 2016, pp. 191–198.

[160] L. Deng, J. Li, J.-T. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, *et al.*, "Recent advances in deep learning for speech research at microsoft," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, IEEE, 2013, pp. 8604–8608.

[161] H. Sun, S. Hu, S. McIntosh, and Y. Cao, "Big data trip classification on the new york city taxi and uber sensor network," *Journal of Internet Technology*, vol. 19, no. 2, pp. 591–598, 2018.

[162] H. H. Olsson and J. Bosch, "From opinions to data-driven software r&d: A multi-case study on how to close the'open loop'problem," in *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications*, IEEE, 2014, pp. 9–16.

[163] M. Banko and E. Brill, "Scaling to very very large corpora for natural language disambiguation," in *Proceedings of the 39th annual meeting on association for computational linguistics*, Association for Computational Linguistics, 2001, pp. 26–33.

[164] E. Deelman and A. Chervenak, "Data management challenges of data-intensive scientific workflows," in *2008 Eighth IEEE International Symposium on Cluster Computing and the Grid (CCGRID)*, IEEE, 2008, pp. 687–692.

[165] P. Vassiliadis, "A survey of extract–transform–load technology," *International Journal of Data Warehousing and Mining (IJDWM)*, vol. 5, no. 3, pp. 1–27, 2009.

[166] J. Trujillo and S. Luján-Mora, "A uml based approach for modeling etl processes in data warehouses," in *International Conference on Conceptual Modeling*, Springer, 2003, pp. 307–320.

[167] T. Rabl and H.-A. Jacobsen, "Big data generation," in *Specifying Big Data Benchmarks*, Springer, 2012, pp. 20–27.

[168] S. H. Khandkar, "Open coding," *University of Calgary*, vol. 23, p. 2009, 2009.

[169] M. Shepperd, "Data quality: Cinderella at the software metrics ball?" In *Proceedings of the 2nd International Workshop on Emerging Trends in Software Metrics*, 2011, pp. 1–4.

[170] X. Chu, I. F. Ilyas, S. Krishnan, and J. Wang, "Data cleaning: Overview and emerging challenges," in *Proceedings of the 2016 International Conference on Management of Data*, 2016, pp. 2201–2206.

[171] E. Rahm and H. H. Do, "Data cleaning: Problems and current approaches," *IEEE Data Eng. Bull.*, vol. 23, no. 4, pp. 3–13, 2000.

[172] A. Zhang, S. Song, and J. Wang, "Sequential data cleaning: A statistical approach," in *Proceedings of the 2016 International Conference on Management of Data*, 2016, pp. 909–924.

[173] A. R. Munappy, D. I. Mattos, J. Bosch, H. H. Olsson, and A. Dakkak, "From ad-hoc data analytics to dataops," in *Proceedings of the International Conference on Software and System Processes*, 2020, pp. 165–174.

[174] M. Botha, A. Botha, and M. Herselman, "On the prioritization of data quality challenges in e-health systems in south africa," in *Proceedings of the 2015 Annual Research Conference on South African Institute of Computer Scientists and Information Technologists*, 2015, pp. 1–10.

[175] T. Botsis, G. Hartvigsen, F. Chen, and C. Weng, "Secondary use of ehr: Data quality issues and informatics opportunities," *Summit on Translational Bioinformatics*, vol. 2010, p. 1, 2010.

[176] K. Sha and S. Zeadally, "Data quality challenges in cyber-physical systems," *Journal of Data and Information Quality (JDIQ)*, vol. 6, no. 2-3, pp. 1–4, 2015.

[177] I. Ryu, "Issues and challenges in developing embedded software for information appliances and telecommunication terminals," in *Proceedings of the ACM SIGPLAN 1999 workshop on Languages, compilers, and tools for embedded systems*, 1999, pp. 104–120.

[178] A. Kumar, M. Boehm, and J. Yang, "Data management in machine learning: Challenges, techniques, and systems," in *Proceedings of the 2017 ACM International Conference on Management of Data*, 2017, pp. 1717–1722.

[179] A. Raj, J. Bosch, H. H. Olsson, and T. J. Wang, "Modelling data pipelines," in *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, IEEE, 2020, pp. 13–20.

[180] T. Kosar, G. Kola, and M. Livny, "Data pipelines: Enabling large scale multi-protocol data transfers," in *Proceedings of the 2nd Workshop on Middleware for Grid Computing*, 2004, pp. 63–68.

[181] D. Baylor, E. Breck, H.-T. Cheng, N. Fiedel, C. Y. Foo, Z. Haque, S. Haykal, M. Ispir, V. Jain, L. Koc, *et al.*, "Tfx: A tensorflow-based production-scale machine learning platform," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017, pp. 1387–1395.