MALMÖ UNIVERSITY STUDIES IN COMPUTER SCIENCE NO 17, LICENTIATE THESIS

MEENU MARY JOHN

DESIGN METHODS AND PROCESSES FOR ML/DL MODELS

MALMÖ UNIVERSITY 2021

MEENU MARY JOHN

DESIGN METHODS AND PROCESSES FOR ML/DL MODELS

ISBN 978-91-7877-198-1 (print)
ISBN 978-91-7877-199-8 (pdf)

MALMÖ UNIVERSITY
205 06 MALMÖ, SWEDEN
WWW.MAU.SE

MALMÖ UNIVERSITY

DESIGN METHODS AND PROCESSES FOR ML/DL MODELS

Malmö University,
Studies in Computer Science No 17,
Licentiate Thesis

ISBN 978-91-7877-198-1 (print)
ISBN 978-91-7877-199-8 (pdf)
Holmbergs, Malmö 2021

MEENU MARY JOHN

DESIGN METHODS AND PROCESSES FOR ML/DL MODELS

Malmö University, 2021
Faculty of Technology and Society
Department of Computer Science and Media Technology

Studies in Computer Science

Faculty of Technology and Society
Malmö University

1. Jevinger, Åse. *Toward intelligent goods: characteristics, architectures and applications*, 2014, Doctoral dissertation.
2. Dahlskog, Steve. *Patterns and procedural content generation in digital games: automatic level generation for digital games using game design patterns*, 2016, Doctoral dissertation.
3. Fabijan, Aleksander. *Developing the right features: the role and impact of customer and product data in software product development*, 2016, Licentiate thesis.
4. Paraschakis, Dimitris. *Algorithmic and ethical aspects of recommender systems in e-commerce*, 2018, Licentiate thesis.
5. Hajinasab, Banafsheh. *A Dynamic Approach to Multi Agent Based Simulation in Urban Transportation Planning*, 2018, Doctoral dissertation.
6. Fabijan, Aleksander. *Data-Driven Software Development at Large Scale*, 2018, Doctoral dissertation.
7. Bugeja, Joseph. *Smart Connected Homes: Concepts, Risks, and Challenges*, 2018, Licentiate thesis.
8. Alkhabbas, Fahed. *Towards Emergent Configurations in the Internet of Things*, 2018, Licentiate thesis.
9. Paraschakis, Dimitris. *Sociotechnical Aspects of Automated Recommendations: Algorithms, Ethics, and Evaluation*, 2020, Doctoral dissertation.
10. Tegen, Agnes. *Approaches to Interactive Online Machine Learning*, 2020, Licentiate thesis.
11. Alvarez, Alberto. *Exploring the Dynamic Properties of Interaction in Mixed-Initiative Procedural Content Generation*, 2020, Licentiate thesis.
12. Alkhabbas, Fahed. *Realizing Emergent Configurations in the Internet of Things*, 2020, Doctoral dissertation.
13. Ashouri, Majid. *Towards Supporting IoT System Designers in Edge Computing Deployment Decisions*, 2021, Licentiate thesis.
14. Bugeja, Joseph. *On Privacy and Security in Smart Connected Homes*, 2021, Doctoral dissertation.
15. Azadvar, Ahmad. *Predictive Psychological Player Profiling*, 2021, Licentiate thesis.
16. Holmberg, Lars. Human in Command Machine Learning, 2021, Licentiate thesis.
17. John, Meenu Mary. Design Methods and Processes for ML/DL Models, 2021, Licentiate thesis.

Dedicated to Ephraim Joseph

ABSTRACT

Context: With the advent of Machine Learning (ML) and especially Deep Learning (DL) technology, companies are increasingly using Artificial Intelligence (AI) in systems, along with electronics and software. Nevertheless, the end-to-end process of developing, deploying and evolving ML and DL models in companies brings some challenges related to the design and scaling of these models. For example, access to and availability of data is often challenging, and activities such as collecting, cleaning, preprocessing, and storing data, as well as training, deploying and monitoring the model(s) are complex. Regardless of the level of expertise and/or access to data scientists, companies in all embedded systems domain struggle to build high-performing models due to a lack of established and systematic design methods and processes.

Objective: The overall objective is to establish systematic and structured design methods and processes for the end-to-end process of developing, deploying and successfully evolving ML/DL models.

Method: To achieve the objective, we conducted our research in close collaboration with companies in the embedded systems domain using different empirical research methods such as case study, action research and literature review.

Results and Conclusions: This research provides six main results: First, it identifies the activities that companies undertake in parallel to develop, deploy and evolve ML/DL models, and the challenges associated with them. Second, it presents a conceptual framework for the continuous delivery of ML/DL models to accelerate AI-driven business in companies. Third, it presents a framework based on current literature to accelerate the end-to-end deployment process and advance knowledge on how to integrate, deploy and operationalize ML/DL models. Fourth, it develops a generic framework with five architectural alternatives for deploying ML/DL mod-

els at the edge. These architectural alternatives range from a centralized architecture that prioritizes (re)training in the cloud to a decentralized architecture that prioritizes (re)training at the edge. Fifth, it identifies key factors to help companies decide which architecture to choose for deploying ML/DL models. Finally, it explores how MLOps, as a practice that brings together data scientist teams and operations, ensures the continuous delivery and evolution of models.

Keywords. Machine Learning, Deep Learning, Development, Deployment, Evolution

LIST OF PUBLICATIONS

This thesis is based on the following publications:

[1] M. M. John, H. H. Olsson, and J. Bosch, "Developing ML/DL Models: A Design Framework," in *Proceedings of the International Conference on Software and System Processes*, pp. 1–10, 2020.

[2] M. M. John, H. H. Olsson, and J. Bosch, "AI on the Edge: Architectural Alternatives," in *46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pp. 21–28, IEEE, 2020.

[3] M. M. John, H. H. Olsson, and J. Bosch, "AI Deployment Architecture: Multi-case Study for Key Factor Identification," in *27th Asia-Pacific Software Engineering Conference (APSEC)*, pp. 395–404, IEEE, 2020.

[4] M. M. John, H. H. Olsson, and J. Bosch, "Architecting AI Deployment: A Systematic Review of State-of-the-art and State-of-practice Literature," in *International Conference on Software Business*, pp. 14–29, Springer, 2020.

[5] M. M. John, H. H. Olsson, and J. Bosch, "Towards MLOps: A Framework and Maturity Model," in *To appear at the 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, IEEE, 2021.

[6] M. M. John, H. H. Olsson, and J. Bosch, "Towards an AI-driven Business Development Framework," *To appear at the ICSSP Special issue of the Journal of Software: Evolution and Process*, 2021.

PERSONAL CONTRIBUTION

My contribution to the publications for which I am first author is listed below, using the CRediT (Contributor Roles Taxonomy) author statement [1]:

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

ACKNOWLEDGEMENT

First and foremost, I would like to thank my main supervisor Helena Holmström Olsson for her motivation, support and guidance. Although she has contributed to my research in different ways, I would like to highlight two in particular. First, it is her expertise in formulating research questions and selecting appropriate research methods. Second, her insights and ideas on conducting interviews and analyzing data. Her continuous availability is a great help in designing my thesis and research path.

Secondly, I would like to thank Jan Bosch, my co-supervisor, for sharing his technical knowledge and engineering mindset to guide my research. I also appreciate his continuous critique of my research, which is never short of suggestions for improvement.

Thirdly, I would like to thank my examiner - Jan Persson and the follow-up group members - Paul Davidsson and Romina Spalazzese, for their academic guidance during the licentiate and ISP process. Their suggestions and discussions on my study plan influenced several decisions I made in this research. I would like to thank Henritte Lucander for helping me plan my teaching schedule and Sussane Lundborg for assisting me with the licentiate and ISP procedures.

I would also like to thank all my fellow PhD students for discussions and fikas, especially Alberto Alvarez, Lars Holmberg, Sergei Dytckov and Johan Salo.

Last but not least, I would like to thank the companies in Software Center with whom I was privileged to work closely during this research. The practitioners in the companies have always supported and helped me. Their input has been instrumental in advancing my research.

CONTENT

1 Introduction

Throughout history, technological innovation has been the primary driver of economic growth. Companies use these technological innovation to transform their core processes and business models, increase value delivery and accelerate innovation. Due to high data availability and tremendous advances in high-performance parallel hardware such as GPUs and FPGAs [2] [3], Artificial Intelligence (AI) and especially Machine Learning (ML) and Deep Learning (DL) are increasingly adopted in embedded systems companies. ML/DL technologies are used in sectors such as telecommunications, automotive [4], finance [5] [6], defense, healthcare [7] [8], manufacturing, etc. and various other sectors.

Although most companies have ongoing AI initiatives, previous research shows that the transition from prototype to production-ready deployment of ML/DL models is challenging and most companies are struggling with [9] [10] [11]. According to a study by Gartner [12], 47% of AI projects remain in the prototype phase because of a lack of tools and processes to develop and manage a production-ready AI system. As previous research has shown, companies face challenges in developing, deploying, and evolving ML/DL models [10] [13] [14] [15] [9] [16] in practice.

Software Engineering (SE) is undergoing a major transformation with the introduction of ML/DL technology [17]. Engineering ML/DL systems in real-world environments is challenging because they add extra complexity to the development of traditional software. According to [18], there are three fundamental differences in the construction of ML models compared to previous technologies. First, in the case of ML, additional effort is required to find, generate, manage and version data compared to software code. Second, in addition to their SE skills, teams need sufficiently deep knowledge of ML to build, evaluate and tune models from scratch [13]. Third, it is more difficult to maintain strict module boundaries between ML models than it is for SE modules. These differences also apply to DL

models. From these differences, it is clear that methods, processes, and techniques designed for the development of traditional software systems and applications are less effectively applied to the development, deployment and evolution of ML/DL models. To address this problem, recent evidence [10] shows that established SE principles, processes, approaches and methods [19] [20] need to be considered and extended in the development, deployment and evolution of ML/DL systems.

Despite numerous promising examples, there is a schism in the ML, DL and SE communities. The ML and DL communities are concerned with algorithms and their performance, while the SE community is concerned with the implementation and deployment of software-intensive systems. In this context, it is crucial to bring together the knowledge and experience of these two communities.

Previous research shows that even experienced practitioners have difficulty developing, deploying and evolving the ML/DL models. According to [13], the development of ML requires the skills of ”high priests”. To support this assertion, [21] suggested that ”crafting these solutions generally requires knowledge possessed by few”. Data scientists involved in the development, deployment and evolution of ML /DL models often have a broad range of skills in areas such as mathematics and statistics, ML, DL and AI, databases, cloud computing, and data visualization [22]. Even when companies are using ML/DL technology extensively in embedded systems, they struggle to recruit adequate data scientists [23]. Therefore, when data science teams are formed in a company, practitioners with a background in SE or business analytics, data governance and ethics experts usually take on the role of data scientist [22] [24]. Based on a survey of 16,716 practitioners (15 job titles) working in data science conducted by Kaggle [25], most respondents were ”data scientists” (15%), followed by ”engineer” (14%), ”researcher” (10%), ”data analyst” (7%), ”business analyst” (5%) and the remaining categories (5%). Data science employs a large number of practitioners in a variety of job categories and titles, and its diversity increases over time [26]. Data scientists use a variety of tools, ranging from simple programming languages to shared notebooks to graphical canvases, to set data pipeline [25] and apply different skills ranging from data wrangling to modeling to visualization [27] [28]. Despite the fact that data science offers powerful tools for gaining insights, practitioners' knowledge of how to apply these tools to tailored situations is limited. On the other hand, while small companies have easy access

to local and domain knowledge, they lack the skills to apply data science tools to their problems [29]. According to the HCI (Human-computer interaction) and ML literature [21], people involved in the development of ML models can be divided into three categories: a) Experts, b) Intermediate users of ML (some experience with ML, but without the deep knowledge that experts possess) [30] and c) Amateurs (interact with ML systems without ML knowledge, such as some domain experts and end users). In summary, ongoing research and surveys show that practitioners have difficulty operationalizing and standardizing work processes. On the other hand, it also shows that companies across the domain, regardless of the level of expertise [13]and/or access to data scientists, struggle to create high-performing models due to a lack of established and systematic design methods and processes.

To address these issues, we focus our research on two main problems: a) the challenge of moving from prototype to deployment of ML/DL models and b) the lack of data science expertise (and thus the need for others to contribute to the development, deployment and evolution of ML/DL models). Our research is motivated by the fact that there is a growing interest in establishing sound SE methods and practices in the development of ML and DL systems [18]. To do this, it is necessary to consider and adapt well-established SE practices that have been largely ignored or have had a narrow focus in the ML literature [13] [15]. Therefore, we focus on developing systematic and structured design methods and processes for model development, deployment, and evolution to not only empower and encourage experienced data scientists, but also enable less experienced data scientists, software engineers and even non-experts to effectively approach the process of developing, deploying, and evolving ML/DL models in today's embedded systems domain where there is a shortage of highly skilled data scientists.

To achieve this, we are conducting studies that cover the end-to-end process of developing ML/DL models as well as in-depth studies that focus on the specifics of the deployment phase (architectural alternatives for deploying AI at the edge and the key factors that lead to the selection of a particular architecture) and how these models can be operationalized in large-scale embedded systems and how ML systems can be continuously delivered and evolved using MLOps (Machine Learning Operations). This research is conducted in collaboration with AI-enhanced companies in embedded systems domain, using different research approaches and tech-

niques.

This thesis is organized as follows: Chapter 1 presents the need for systematic and structured design methods and processes for the development, deployment and evolution of ML/DL models. Chapter 2 provides the background to the study. Chapter 3 describes the research methodology and process employed in the study. Chapters 4, 5, 6, 7, 8 and 9 are based on the publications. Chapter 10 concludes the thesis with a discussion of the main findings and future work.

2 Background

This section provides background information and describes related work in this thesis. Section 2.1 discusses contemporary SE and the different approaches to software development. Section 2.2 explains the continuous practices that emerge in response to the dynamic environment in which SE takes place. Section 2.3 helps to understand the specifics of ML and DL. Section 2.4 presents workflows and practices for the development, deployment, and evolution of ML/DL models. Section 2.5 provides a brief overview of the challenges reported and faced by practitioners.

2.1 Contemporary Software Engineering

Software-intensive companies have experienced a paradigm shift over the past decade due to the digitalization of products and services [31]. According to Gartner [32], digitalization is ..."the use of digital technologies to change a business model and provide new revenue and value-producing opportunities; it is the process of moving to a digital business". This paradigm shift driven by digitalization has a significant impact not only on the products and services that companies produce, but also on the way these products and services are produced, i.e. on the development approaches themselves. If companies want to remain competitive, they must understand, embrace and maximize the benefits of different development approaches. As software becomes an integral part and a key differentiator, companies are being forced to transform their business to enable them to develop software at large-scale [33] [34] and also to drive the approaches they use to develop these systems to ensure value for their customers. With the increasing availability of and access to data and the advent of AI and technologies such as ML and DL, companies are complementing their traditional requirements engineering approaches to development with other approaches [31].

However, when developing software to be used as part of an embedded system, there are a number of factors to consider compared to pure software companies [33] [34]. As shown in [31], there are three different development approaches to contemporary software development (1) Requirement driven development, (2) Outcome/data driven development and (3) AI driven development. In requirement driven development approach, the software is built according to the specification. This approach is used when the new features or functions are fully understood and specified and business revenue does not depend on frequent release of new features. Requirements are gathered, specified, and carefully documented as the primary input to development teams and as a mechanism to ensure that a system is built and delivered in accordance with customer preferences and needs, especially in embedded systems companies [35] [36].

In the outcome/data driven development approach, development teams are given a quantitative target to achieve and are asked to experiment with different solutions to improve the metric. This approach is primarily used for developing new features that are used by customers on a regular basis and for innovation initiatives when there is uncertainty about how to implement a new feature. This approach is used in the embedded systems domain where experimentation with different software solutions to determine and validate customer value is becoming increasingly important [37] [38] [39].

The third and rapidly emerging new approach to software development is AI driven development [31]. In this approach, companies have access to large data sets and use AI techniques such as ML and DL to develop components that provide insights based on input data and learn from past actions, with the goal of minimizing prediction errors. An example of this type of development is object recognition in automobiles. This development approach is used when manual processing of the software would be either too difficult, time consuming or expensive. In this approach, the computer is given the task of solving a problem. Problems arise when building a production-ready AI system in a company that lacks mechanisms and infrastructure for conducting experiments has limited resources for large and complex data sets, and whose corporate culture, skills, and interests are incompatible with cross-functional collaboration.

As AI driven development is still in its infancy in many companies, they face challenges related to the development, deployment and evolution of ML/DL models. This highlights the need to define systematic and

structured design methods for developing, deploying and evolving ML/DL models.

2.2 Continuous Development Practices

Regardless of software, data or ML/DL components, companies are looking for continuous development practices that enable ongoing development as well as delivery of these components [40] as more opportunities exist to deliver improvements to customers on a regular basis. Companies are adopting DevOps, DataOps, and MLOps as practices that go beyond agile methods and represent new ways of working for software-intensive embedded systems. They are using these practices to improve automation and quality of production in terms of development, data and ML operations.

2.2.1 DevOps

DevOps stands for development and operations [41]. It aims to ”reduce the time between committing a change to a system and the change being placed into normal production while ensuring high quality” [42]. The goal is to merge development, quality assurance and operations into a single continuous set of processes.

2.2.2 DataOps

According to reference [43], DataOps can be defined as "an approach that accelerates the delivery of high-quality results by automation and orchestration of data life cycle stages". It brings speed and agility to the end-to-end process of data pipelines, from collection to delivery.

2.2.3 MLOps

Following continuous practices in SE (DevOps and DataOps), companies are trying to enable continuous practices in rapid deployment of ML models as well [44]. The practice of continuous delivery of ML is referred to as MLOps. It is an engineering practice that applies DevOps principles to ML systems to unify the development and operation of the ML system. In practice, ML models are embedded in a larger software system that hosts, provides access to and monitors the ML features [45]. Because the model is often only a small component of the overall software system,

the interaction of the model with the rest of the software and context is extremely critical [15]. MLOps helps data scientists and engineers generate long-term value and reduce risk in data science, ML and AI initiatives through standardization and streamlining Machine Leaning Lifecycle [46]. MLOps enables data scientists to collaborate and increase the speed of delivery and quality of model development by monitoring, validating, and driving machine learning models. This is similar to the way DevOps helps software engineers develop, test and deploy software faster and with fewer defects. MLOps supports the data science life cycle in the same way that DevOps supports the application development lifecycle [47].

2.3 Machine Learning (ML)/Deep Learning (DL)

Companies that integrate ML/DL into their software-intensive systems seek to gain an increasingly competitive advantage over companies that do not use ML/DL technologies. The term ML was coined by Arthur Samuel [48]. According to Tom M. Mitchell [49], ML can be defined as ”a computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E”. ML consists of four main learning paradigms, namely supervised learning (SL), unsupervised learning (UL), semi-supervised learning (SSL) and reinforcement learning (RL) [50] [51]. Unsupervised learning attempts to find hidden structures in labeled data while supervised learning infers a function from labeled training data. SSL presents a challenging learning setting where models are constructed from training data typically consisting of a small number of labeled instances and a large number of unlabeled instances. Reinforcement learning seeks to maximize reward based on actions performed in the environment. ML has rapidly evolved from a laboratory curiosity to a practical and commercially used technology.

Deep Learning (DL), a subset of ML builds on multiple levels of representation by composing simple but non-linear modules. According to Yoshua Bengio [52] ”Deep learning algorithms seek to exploit the unknown structure in the input distribution in order to discover good representations, often at multiple levels, with higher-level learned features defined in terms of lower-level features”. Compared to conventional ML algorithms, DL methods are very effective in exploring high-dimensional data.

2.4 ML/DL Workflows and Practices

Previous research has identified workflows and practices for developing, deploying and evolving models.

According to [53], the workflow of ML consists of three components: a) Data Pre -processing (DPR), b) Learning/Inference (L/I) and c) Post Processing (PPR). In DPR, the raw data is converted into a format compatible with the ML algorithm through data manipulations such as data cleaning and feature extraction. In L/I, the learned ML model based on the transformed data is used to make predictions for unseen data. PPR deals with the operations after performing inference based on a learned model. Companies that integrate ML workflow into the software development process employ polymath data scientists on their software teams [18]. As ML becomes more prevalent, data scientists must specialize as domain experts who have a deep understanding of the business case, modelers who develop predictive models and platform developers who are responsible for creating cloud-based infrastructure.

The ML workflow used by Microsoft [18] to develop AI applications includes nine stages with feedback loops. They are model requirements, data collection, data cleaning, data labeling, feature engineering, model training, model evaluation, model deployment and model monitoring. In model requirements, practitioners decide on the features and types of models that are appropriate for a particular business problem. In later stages, they find and collect data, remove noisy data and assign ground labels. After labeling the data, they perform feature engineering, train and tune the models, and evaluate the model against predefined-metrics. The model for inference is deployed on target devices and continuously monitored for performance degradation. This workflow is illustrated in various forms in [54] [13] [55] [56]. These workflows have similarities to KDD [57] and CRISP-DM [58] in the context of data science and data mining.

2.5 Challenges in Developing, Deploying and Evolving ML/DL Models

Despite well-established workflows and practices as described above, the development of ML/DL models suffers from a number of challenges. Reference [10] attempts to identify and classify various SE challenges that exist

when building and deploying ML models in software-intensive systems. Experts encounter problems in building systems with ML algorithms, as described in [13] [14]. The development of ML models by non-experts is explored in [21] using their experience, knowledge and blind spots. The unique potentials and pitfalls of non-experts are highlighted in this paper. In [15], several ML specific risks are illustrated that should be considered in system design using the SE framework of technical debt. These include boundary erosion, hidden feedback loops, data dependencies, configuration debt, changes in external world, etc. Many studies have been conducted to test software [59] and ML models. However, studies that consider the combination of SE and ML have not been as extensively researched [60]. Implementing a production-ready ML system needs to be tested, as does the software [19]. The challenges in the intersection of SE and ML can be grouped into development, production, and organizational challenges [9]. DL techniques require special infrastructure support compared to ML and traditional software development [61]. DL requires high-end infrastructure to train large amount of data in reasonable time and achieve better accuracy.

3 Research Methodology & Process

Software engineering is a multidisciplinary field that spans various social and technological boundaries. Therefore, it has become imperative to investigate not only the tools and processes used by software engineers, but also the social and cognitive processes that surround them and in which software and systems development takes place, in order to better understand how individual software engineers create and maintain complex, evolving software systems and how teams and organizations coordinate their efforts [62]. Research methodology is a scientific and systematic way for solving a research problem. In research methodology, ontological and epistemological principles are translated into guidelines that indicate how research should be conducted and into principles, procedures and practices that govern research [63]. To conduct successful research, one must understand not only research methods and techniques, but also methodology.

There are two strands of research methodology: (a) Qualitative research and (b) Quantitative research. The aim of qualitative research is "Development of concepts which help us to understand social phenomena in natural (rather than experimental) settings, given due emphasis to the meanings, experiences and views of the participants" [64]. This research methodology helps to provide richer information, deeper insights into the phenomenon under study and an understanding of the perceptions that underlie and influence the various negative impacts studied. The goal of quantitative research is to "explain behavior in terms of specific causes (independent variables) and the measurement of the effects of those causes (dependent variables)". Quantitative research improves the generalizations of a larger number of subjects and thus achieves greater objectivity. We adopted a qualitative approach to our research.

Previous studies have shown little empirical support for the development, deployment and evolution of ML/DL models. The end-to-end process of developing, deploying, and evolving models and integrating them into

larger embedded systems presents a significant challenge. This is especially true for large, complex, highly regulated and safety-critical embedded systems in areas such as telecommunications, automotive, defence, security, healthcare, etc. With this in mind, the goal of this research is to enable not only experienced data scientists, but also less experienced data scientists, software engineers, and even non-experts to address and promote the development, deployment, and evolution of ML/DL models.

This chapter is organized into sections covering the research questions, qualitative research, case companies involved in the research, research methods and research techniques.

3.1 Research Questions

The objective of this research is to develop systematic and structured design methods and processes to support the development, deployment, and evolution of ML/DL models. To achieve the research objective, we have adopted a qualitative research methodology. The research focuses on three primary research questions.

RQ1: What are the activities performed and challenges experienced in the process of **developing** ML/DL models in companies with software-intensive embedded systems?

RQ2: What are the best practices, challenges encountered and architectural choices made by practitioners in **deploying and operationalizing** ML/DL models?

RQ3: How can MLOps practices help large-scale embedded systems companies achieve continuous **delivery and evolution** of ML/DL models?

The first research question (RQ1) aims to identify the phases and activities that companies undertake simultaneously to develop, deploy and evolve models and the challenges they face. This question also addresses iterations and triggers that optimize model design for better accuracy and identifies the roles and organizational functions involved. The second research question (RQ2) focuses on the specifics of the deployment phase and how these models can be operationalized in large embedded systems. This question aims to improve the understanding of the integration, deployment and operationalization of ML/DL models. For example, it explores

the architectural choices for deploying models at the edge, ranging from centralized cloud to decentralized edge and identifies the factors that affect architectural choices. The third research question (RQ3) analyzes how practices such as MLOps help to ensure the continuous development and delivery of ML/DL models. This question aims to provide practitioners with a blueprint on how to effectively implement continuous practices, i.e. MLOps, into the company's business.

3.2 Research Context

The research reported in this thesis was conducted as part of Software Center [65]. Software Center is a large research collaboration consisting of seventeen companies in the embedded systems domain and five Swedish universities. It initiates and carries out research projects with industrial and academic partners in an active, close and long-term collaboration. As the name suggests, the companies involved in this collaboration develop products and services based solely on or supported by software. As best practices change rapidly in SE, Software Center enables companies to innovate and build new core capabilities to increase their competitiveness by collaborating to develop novel software engineering as a core competency. In addition, these companies are incorporating ML/DL models into their software-intensive embedded systems to provide value to their customers.

Each calendar year, the Software Center organizes an ongoing collaboration between university researchers and practitioners from partner companies in two six-month sprints. During a typical sprint, practitioners and researchers with similar interests meet frequently in company-specific and cross-company workshops to identify current research challenges, advance ongoing research, coordinate work and present and validate results. Each sprint begins with a joint "kick-off" workshop session where practitioners and researchers agree on a research focus and frequent research activities. The sprint ends with a reporting workshop attended by all partner companies. In addition, the Software Center day is held once a year where all partner companies and universities come together to discuss the recent research and industry challenges. At the end of each sprint, the research proposal for the next sprint is presented to the steering committee (selected practitioners from each partner company) for approval. Companies engage in proposals that they consider interesting and relevant to their specific domain. On the other hand, they wish to remain anonymous throughout the

research as we reflect on the challenges, pitfalls and limitations in their current practice of developing, deploying and evolving ML/DL models. Based on their engagement in our research, we categorize ten of the seventeen companies as primary and the rest as secondary. Primary companies are those with whom we worked continuously and very closely and who actively contributed to the research during the sprints. Secondary companies are those that did not play an active role in the research during the sprint, but did participate in activities where we shared our results.

3.2.1 Primary Case Companies

Below, we present the primary case companies contributed to this research:

Table 1: Primary Case Companies

Case	Description
A	A multinational telecommunication company providing equipment and telecommunications systems to mobile and fixed-line operators, communications networks and multimedia solutions
B	An automotive company that manufactures and supplies transport solutions, construction technologies and commercial vehicles
C	A multinational packaging company that provides processing and packaging solutions for food and beverage products
D	Functions as an innovation center for an automotive company that introduces smarter solutions for passenger vehicles
E	Manufactures and distributes pumps for heating, air conditioning and fluid handling
F	Develops software and hardware for various products, from home appliances to vehicles. They have standards to ensure product design, functionality and quality
G	An automotive manufacturer that uses AI to develop autonomous driving technology
H	A multinational company that provides systems for generating and transmitting power and medical diagnostics
I	A global provider of transportation solutions
J	Provides a platform for developing and deploying DL models

3.3 Qualitative Research

To answer the research questions of interest, we used a qualitative research approach. According to [66], qualitative research ”involves an interpretive, naturalistic approach to the world. This means that qualitative researchers study things in their natural settings, attempting to make sense of, or interpret phenomena in terms of the meanings people bring to them”. Qualitative research examines behaviours, beliefs and motivations in a particular context. This involves collecting, analysing and interpreting data that cannot simply be reduced to numbers [67]. The main purpose of qualitative research is to answer the following questions: ”Why?, How?, What is the process? What are the influences or context?” [68]. Qualitative research explores people's experiences in depth using research techniques such as interviews, group interviews (focus groups), observations and documents [69]. We chose qualitative research as a mode of inquiry [70] and used interviews, group interviews (focus groups) and observations to study real-world environments. Qualitative research is appropriate in our study to understand how a purposively selected company [68] or practitioners within that company perceive the development, deployment and evolution of ML/DL models.

3.3.1 Case Study Research

Yin defines a case study [71] ”as an empirical inquiry that investigates a contemporary phenomenon within its real-life context, especially when the boundaries between phenomenon and context are not clearly evident”. The five steps involved in the case study are as follows [72] (a) Case study design:- establishing the objectives and planning the case study (b) Preparation for data collection:- defining data collection procedures and protocols (c) Collection of evidence:- conducting data collection procedures for the case under study (d) Analysis of collected data:- apply data analysis procedures to the data and (e) Reporting:- packaging the study and its conclusions in reportable formats. According to [73] classification, there are four research purposes: (a) Exploratory, which helps to uncover processes, gain new insights and develop ideas and hypotheses for new research, (b) Descriptive, which describes a situation or phenomenon, (c) Explanatory, which seeks an explanation for a situation or problem that is not necessarily in the form of a causal relationship and (d) Improving, which seeks to improve some aspect of the phenomenon under study. The

selection of an appropriate case is a crucial part of case study research. It uses purposive sampling to choose relevant cases for the study [62]. Depending on the phenomenon under study, a case study can be single or multiple. A multiple case study allows for a deeper understanding of each case as a unit by comparing similarities and differences. Case study research also allows for the study of a phenomenon over a longer period of time, which is called a longitudinal case study [72]. Case studies are conducted not only to increase knowledge about individuals, groups and organizations [74], but also to bring about change in the phenomena under study based on the results of case studies. In SE, case studies are used to better understand how and why SE was undertaken, thus improving the SE process and the resulting software products [72].

We chose both an exploratory and a longitudinal multiple case study to understand, clarify or demonstrate new technique capabilities, methods, tools, processes, technologies or organizational structures, and to focus on an in-depth investigation of a contemporary phenomenon that is difficult to study separately in its real-world context. Through the multiple case study, we explored the different phases, activities performed by practitioners and associated challenges of developing ML/DL models, the factors influencing the selection of an architecture for deploying AI models, and the ways in which companies adopt MLOps to ensure continuous model delivery.

3.3.2 Action Research

Action research [75] is a research method in which researchers and practitioners work closely together to solve a research problem in a company. Action research can be considered ”the most realistic research setting found because the setting of the study is the same as the setting in which the results will be applied for a given organization, apart from the presence of the researchers” [76] and its application reinforces ”more on what practitioners do than what they say they do” [77]. The action research cycle consists of five steps [78]: (1) Diagnosing - focusing on understanding, describing and agreeing on a problem to be solved, (2) action planning and design - focusing on discovering alternatives to solve a problem and selecting an appropriate choice, (3) action taking - focusing on implementing the action plan, (4) Evaluation - capturing the impact of the action using a variety of data collection methods and (5) Specifying learning

- determining learnings based on the evaluation. According to [79] [78], the entire process is documented using notes, semi-structured interviews, questionnaires and audio recordings. In addition, several versions of software prototypes serve as documentation of changes made during the action research cycles. Action research can be characterized [80] as intervention-focused, iterative and participatory. In intervention-focused research, the researcher introduces the action (intervention) to understand its consequences in the context under study. In iterative research, the implemented action (e.g., a method or tool) is iteratively improved through several cycles of action research [81] [78]. In participatory research, the researcher is given the opportunity to actively participate in the implementation of actions and to observe the context under study. Action research seems to be a well-suited research method to increase the relevance of SE because it emphasizes collaboration between SE researchers and practitioners in real-world settings. It provides a practical benefit to client companies that are trying to solve a real-world problem while exploring the experience of solving the problem [82]. The disadvantage of action research is that the researcher has to invest a lot of time and energy to evaluate the problem in different iterations [62]. In addition, companies often refuse to use their developments as trial cases.

Action research is a variant of case study research. Its main goal is to "influence or change some aspect of what is the focus of the research" [73]. Case study is purely observational whereas action research focuses on and is involved in the change process [83]. Action research is initiated to solve an immediate problem and aims to change the current state and contribute to an improved situation, whereas case studies focus on a specific phenomenon over a longer period of time.

We complemented case study research with action research because we wanted to improve and impact practice. A prerequisite for action research is that both the researcher and the practitioners of the case company work together to identify and agree on a problem to be solved. In our case, the problem to be solved was to develop different architectural alternatives for deploying AI models at the edge. In the next step, the primary researcher spent two days per week for six months at the case company. During this time, the researcher gained an in-depth understanding of the company, its use cases, the technologies used, and the challenges involved. Action research emphasizes participatory research and iteratively develops practical usable solutions. The entire process is documented through interviews,

meetings, discussions, workshops (both individual and group), meeting notes, written diaries and company presentations. As a result of the action research, the researcher developed a generic framework consisting of five architectural alternatives ranging from a centralized to a decentralized approach to deploying AI at the edge. The final results were presented to the case company in the form of a presentation. Based on the action research, we conducted a follow-up study of different case companies to identify the factors that influence the selection of a particular architecture alternative.

3.4 Research Techniques

Research techniques are used to collect empirical data to analyze actions in real industrial contexts [84]. In case study and action research study, we employed a number of different techniques to collect empirical data.

3.4.1 Interviews

Interviews are a data collection technique commonly used in empirical SE research [85]. The main purpose is to collect data about a phenomenon that cannot be captured by quantitative measures. By interviewing practitioners, one gains insight into their work practices, use cases, opinions, thoughts and feelings about a particular topic under study. Based on the number of participants in interviews, they can be divided into (1) Individual interviews, in which one interviewer and one respondent discuss a topic of mutual interest [86] and (2) Group interviews, also known as focus groups [87], in which multiple respondents discuss topics introduced by one or two interviewers. Interviews can be divided into structured, unstructured, and semi-structured interviews. Structured interviews allow for specific research questions as the interviewer has a clear idea of the information to be gathered [69]. In highly structured interviews, responses can be quantified. In unstructured interviews, the interviewer suggests the theme of the interview but has few specific questions in mind [88]. In semi-structured interviews [88], also called focused interviews, specific questions (to obtain predictable information) are combined with open-ended questions (to obtain unexpected types of information). In addition to actually conducting the interviews, several activities are required. These include scheduling appointments with interviewees, gathering background information, preparing interview guides and following up on interviews/meetings, writing summaries and transcribing [85]. In-

terviews can be recorded in a variety of ways, such as with an audio tape recorder or a scribe.

We chose to conduct a semi-structured interview study to gain insights into the activities performed by practitioners and the associated challenges of developing models, to consider the factors influencing the choice of architecture for AI model deployment and to explore how companies adopt MLOps practices.

3.4.2 Observations

Qualitative data collection using observations provides rich insights into the topic under study [62]. Observations define a setting in which a subject performs a task while being observed by an experimenter [89]. An observational approach can be more time consuming for the experimenter and less relaxing for the subject than interviews or questionnaires. Observations are reported on real phenomena that are judged for their interest. They provide valuable clues to practice when findings are not available. They also help to understand an area and lay the foundations for research that will lead to new knowledge or findings over time [90]. Observations can be conducted to study how a particular task is performed by software engineers [62].

Data collection in an observational study can be divided into two subtypes:- observational and inquisitive. Observational data are collected while the process is being carried out, but without interference from the researcher. For example, subjects are asked to think aloud while performing a technique so that the researcher can gain insight into the performance of the technique. Inquisitive data are collected after a process step is completed rather than during its execution. The researcher needs to be more assertive and solicit answers to predefined questions rather than passively observe. For example, at the end of each step, the researcher might ask the subject for qualitative feedback on whether that step was worthwhile or whether the same results could have been better achieved in a different way [89].

Observation can be a first or second degree method. There are many different approaches to observation. One approach is to observe a group of software engineers with a video recorder and analyze the recording later, for example by protocol analysis [91] [92]. Another alternative is to use a

"Think Aloud" protocol, in which the researcher repeatedly asks questions such as "What is your strategy?" and "What are you thinking?" to remind subjects to think aloud. This can be combined with audio and keystroke recordings. Another type is observation in meetings, where participants interact with each other to gain information about the subject of the study. An alternative approach is proposed by [93] where a sampling tool is used to obtain data and feedback from participants.

A particular form of ethnography is participant observation, where the researcher becomes a member of the community under study for a period of time. Here the researcher seeks to understand the community not through the observations of an outsider, but through the privileged perspective that comes from membership. The greatest challenge in ethnographic research is to conduct detailed observations, data collection and analysis while avoiding bias. The researcher needs a high level of training in observational and qualitative data analysis techniques. We used participant observation in our study by participating in stand-up meetings, workshops, interviews and action research in companies.

3.4.3 Multi-vocal Literature Review

A multi-vocal literature review (MLR) aims to identify existing evidence on a particular topic and explore current gaps, suggest areas of investigation and provide a framework for positioning new research activities [94] [95] [96]. An MLR is a form of SLR that includes grey literature as well as published literature. MLRs provide summaries of the state-of- the-art and state-of-practice in a given area. SLRs are conducted according to the procedures summarized by Kitchenham [96] to minimize publication bias and increase the completeness of the search.

We conducted MLRs to explore how ML/DL models are integrated, deployed and operationalized and to examine the MLOps practices used to ensure the continuous delivery and evolution of models. We used semi-structured interviews, workshops and observations to complement the findings of the MLRs in our research. We mapped the case companies to the MLOps maturity model derived using MLRs wth the help of semi-structured interviews, workshops, observations, meetings and stand-up meetings.

3.5 Data Analysis

Qualitative research provides unstructured text-based data. These may be interview transcripts, workshop notes, meeting notes, diary entries, etc. In this thesis, we have used two types of qualitative data analysis: a) Open coding and b) Triangulation.

3.5.1 Open coding

In analysing the qualitative data, we applied elements of open coding to identify key concepts and then group them into conceptual categories [97].

3.5.2 Triangulation

Triangulation increases the precision and validity of our study [98]. Because there were three of us researchers conducting the interviews and reading the interview transcripts, notes and summaries, triangulation allowed for multiple perspectives on the topic under study [99].

3.6 Research Process

The primary step in the research process was to find an idea for the research. The research idea emerged as a result of continuous interactions with Software Center companies that helped us identify current challenges faced by practitioners in AI-enhanced systems in the embedded domain. Of the various challenges listed by practitioners, we wanted to focus primarily on developing a systematic and structured design methodology for the process of developing, deploying and successfully evolving ML/DL models as part of software-intensive and highly complex systems. To achieve this, we propose to conduct studies that cover the end-to-end process of developing ML /DL models, as well as in-depth studies that focus on the specifics of the deployment phase and how to operationalize these models in large-scale embedded systems and how to continuously deliver and evolve Machine Learning Systems (MLOps). The RQs are framed to begin with exploring and identifying the problem and developing and identifying solutions. We have chosen a qualitative research methodology for our thesis.

The overview of the research process is shown in Figure 1. Figure 1 explains the different research methods that were used in the case companies to answer the RQs. It also shows the results obtained to answer the RQs

and the related publications. A, B, C, D, E, F, G, H, I and J denote the case companies included in the research while RQ1, RQ2 and RQ3 denote the RQs.

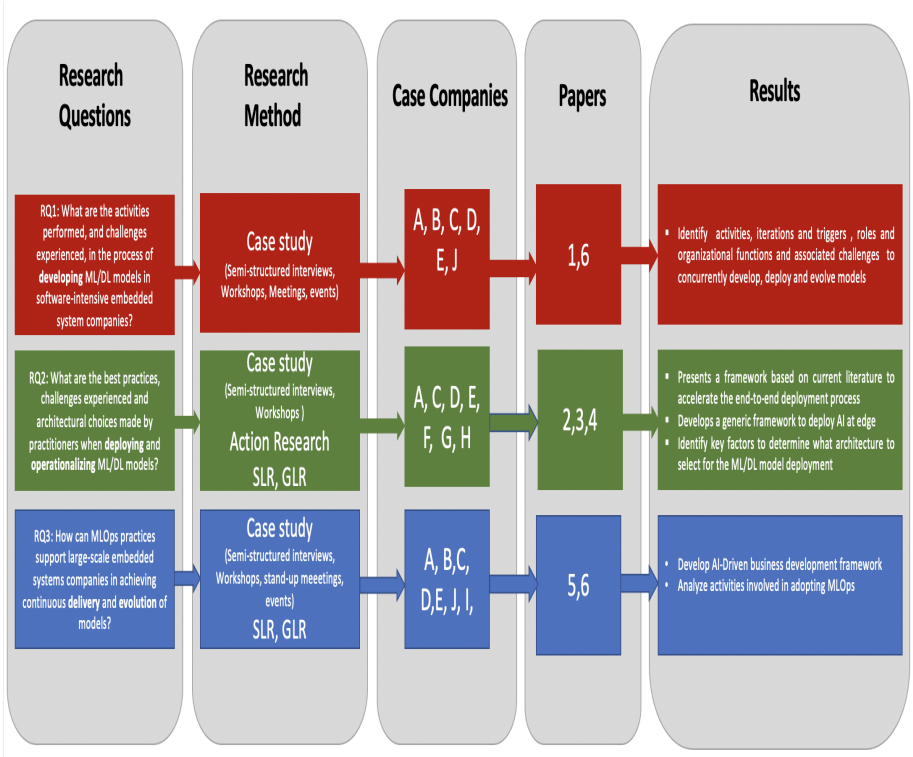

Figure 1: Research Process Overview

The overall research process in this thesis consists of four case studies, one action research study and one literature review and can be described as follows:

3.6.1 Case Study 1: Development of ML/DL Models

Despite the fact that practitioners are interested in developing ML/DL models, they do not follow a systematic and structured approach. In our interviews with practitioners, we asked them "how they deal with the development, deployment and evolution of ML /DL models". Surprisingly, we found that practitioners did not have a clear and concise answer to this question. This has led us to focus our research on this extremely important

topic. In order to investigate the phases of the design process based on the activities performed by practitioners, the challenges encountered and the iterations and triggers between phases to optimize the design process during the development of ML /DL models, we employed an exploratory case study method to three case companies. The case study allows for an in-depth and detailed investigation of the phenomenon based on evidence collected in interviews, workshops and meetings. In case company A, we studied four use cases, while in case companies B and C, we studied only one use case. Semi-structured interviews with open-ended questions are the primary research method used to collect data. We conducted interviews with nine practitioners to gather valuable information. All interviews were scheduled for one hour, two of which were conducted face-to-face with primary researcher and the rest via videoconference. The interviews were transcribed and recorded for further analysis if the interviewees agreed to this. We held five workshops (both cross-company and company-specific) in which nine practitioners discussed the development of ML/DL models and the challenges involved. In addition to the interviews and workshops, we held frequent meetings to learn more about the topic. Through analysis of the interview summary and notes taken during the interviews, workshops and meetings, we used open coding elements to identify key concepts and organize them into conceptual categories. We validated the study through triangulation and sent the findings to the practitioners involved in the study for feedback. Based on the findings, a paper published is presented in reporting workshops, company specific and cross-company workshops, academic lectures and seminars.

3.6.2 Case Study 2: AI-driven Business Development Framework

In this study, we have identified the high-level activities that companies perform in parallel and concurrently to develop, deploy and evolve models. We have also detailed the activities, iterations and triggers used to optimize model design, as well as the roles and organizational functions. We also demonstrated how this study helps companies solve challenges we identified, and discussed different decision points for immediately terminating less valuable business cases. As part of this study, we worked closely with practitioners from six different software-intensive embedded systems companies to examine nine use cases. Of the six case companies,

three companies are involved in our case study 1. Primary data collection techniques include semi-structured interviews, workshops, meetings and events. Eighteen experts from six large companies representing different business areas and domains participated in the interview study. We organized a total of five workshops in which we visited the same case companies to present our preliminary findings based on the study and solicit their feedback. The findings were presented at these meetings and events for validation.

3.6.3 Case Study 3: Key Factor Selection for Ideal Architecture

Although ML/DL is popular, we have found that companies find the transition from prototypes to production-quality models difficult. In previous interactions with companies as part of our action research study, we found that practitioners wrestle with the decision of whether to deploy AI at the edge using a centralized approach, a mix of centralized and decentralized approaches, or a fully decentralized approach. In order to identify key factors as well as prioritization and trade-offs between these key factors that impact the selection of an optimal architecture for AI deployment, we chose to conduct a longitudinal case study. We conduct a case study to obtain in-depth information about the topic under study. This is a continuation of our action research study, working closely with five of the companies previously involved and also including two new companies. We conducted semi-structured interviews with 11 practitioners from 6 case companies and a workshop with thirty practitioners in one of the case companies. Since this is a follow-up study to our action research study, we presented our architectural alternatives to the practitioners at the beginning of the interviews and workshop to give them a better idea. We then asked for their thoughts on the factors relevant to the selection of the specific architecture. The transcripts of the interviews and the notes from the workshops facilitate the collection of our empirical data. To analyze the data, we used investigator triangulation.

3.6.4 Case Study 4: Adoption of MLOps

There is little research on MLOps as it is a recent phenomenon. To improve understanding of how companies practice MLOps, we conduct a SLR and

a GLR to provide a framework that identifies MLOps adoption activities and the stages in which companies evolve as they become more mature and advanced. Following SLR and GLR, we conducted a validation study in three case companies and mapped the case companies to the stages of the maturity model derived from the literature review. For data collection, we used interview studies, workshops, meetings and stand-up meetings in companies. Open coding and triangulation were used for data analysis.

3.6.5 Literature Review 1: End-to-end Deployment Process

Previous studies have shown that very little research has been done on how to manage and deploy models once they have been trained with appropriate ML/DL models. To improve the understanding of how AI models are integrated, deployed, operationalized and evolved, we conducted systematic and grey literature reviews. They were were conducted to obtain an overview of the current state-of-the-art as well as the state-of-practice and voice of practitioners on the end-to-end deployment process. For SLR, we searched IEEE Xplore, ACM Digital Library, Scopus, ScienceDirect and Web of Science. For GLR, we selected articles from companies (filtering .com sites) in PDF format. We present a framework, practices and challenges associated with the end-to-end deployment process based on literature reviews.

3.6.6 Action Research Study 1: Architectural choices for AI Deployment

Based on frequent discussions with practitioners from case company C, a specific problem of interest to both the researcher and the case company was identified for investigation. At this point, the company had already implemented centralized (re)training of models in the cloud with only inference at the edge and they were trying to implement (re)training of models at the edge. To develop a generic framework consisting of different architectural alternatives for deploying AI at the edge, we conducted action research where the researcher worked in close-collaboration with eleven practitioners in the company as study partners. To gain a better understanding of edge/cloud (re)training, inference and transfer learning in the context of four ML/DL use cases, the primary researcher spent two days per week at the company for six months. To drive the research,

we also conducted interviews, meetings, discussions and workshops (both individual and group) and used meeting notes, written diaries and presentations provided by the company. Based on the data collected and working closely with the practitioners, we presented the developed architectural alternatives to the case company and took their feedback. After the architectural alternatives were confirmed by case company C, we conducted a qualitative validation study in four other case companies. The architectural alternatives were first explained to nine practitioners in the four case companies and later their reflections were included. The interviews were scheduled for one hour. Investigator triangulation was used to explore multiple perspectives on the topic. A paper is published based on the findings, which is also presented at reporting workshops, company-specific workshops and cross-company workshops.

Figure 2 shows the companies involved, the research techniques used and the timeline of each research publication that meets the objectives of this thesis. RQ1 is addressed through paper 1 and paper 6. RQ2 through papers 2, 3 and 4. The objectives of RQ3 are achieved through paper 1 and paper 6.

3.7 Threats to Validity

With respect to our thesis, we tried to minimise risks to validity, particularly construct validity, internal validity and external validity.

3.7.1 Construct Validity

Construct validity refers to the extent to which the operational measures studied indicate what the researchers are considering and intend to explore according to the research questions, [83] i.e., whether the theoretical constructs are appropriately interpreted and measured. Some threats to construct validity may affect the results presented in this thesis and to mitigate this risk, different approaches were used depending on the nature of the study.

In our study, to reduce construct validity, a semi-structured interview guide was developed to capture the topic under investigation. A brief description of the topic to be explored is sent to the interviewees before the interview and the topic of the study is explained again during the interview as an introduction. In case of unclear answers, we ask the interviewees to

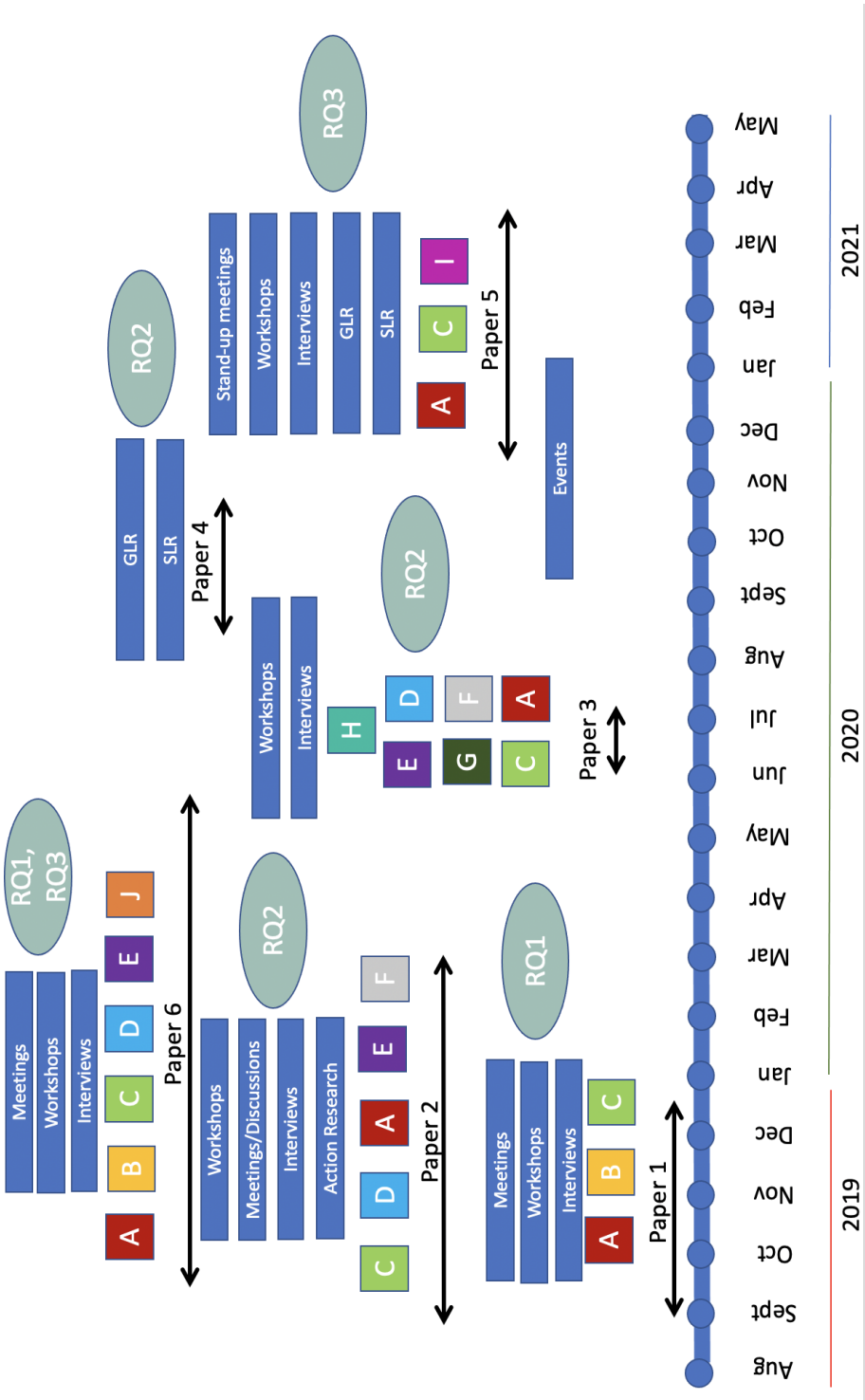

Figure 2: Research Process Timeline

explain the answers.

3.7.2 Internal Validity

Internal validity refers to whether the result is correctly derived from the researcher's conclusions without external factors playing a role. When the researcher investigates whether one factor influences a factor under study, there is a risk that the factor under study is also influenced by a third factor. Yin notes that internal validity primarily concerns studies in which the researcher attempts to explain how and why one event led to another, inferring a causal relationship without considering the presence of other excluded events. Easterbrook et al. [62] further note that it is a common mistake to confuse correlation with causality and that it is much more difficult to prove causality than to show that two variables are correlated.

To reduce internal validity, we presented our study to the practitioners involved/non-involved in the study as well as the two other researchers with extensive knowledge in the field. In addition, the findings were validated by the steering committee of the respective companies.

3.7.3 External Validity

The external aspect of validity is concerned with the extent to which it is possible to generalize the findings of the study and apply them to other situations. Yin defines this as, "An analytic generalization consists of a carefully posed theoretical statement, theory, or theoretical proposition." Easterbrook et al. [62] note that this usually depends on the type of sample used in a study. This idea is echoed by Morse et al. "the sample must be appropriate consisting of participants who best represent or have knowledge of the research topic."

The work presented from cases studied with different teams in different domains is to be seen differently in parts of the company.

3.8 Summary

This chapter has discussed the three RQs that will be addressed in this thesis. The research methods, techniques and designs used to answer the RQs are explained. Table 1 provides an overview of the research methods, techniques and data collection techniques used to answer each RQ. Based

Table 2: Overview of the different studies presented in this thesis

Research Question	Research Objective	Research Method	Technique	Data Collection Technique	Companies and Use cases
1. What are the activities performed, and challenges experienced in the process of developing ML/DL models in software-intensive embedded system companies?	1.Exploration of phases in design process 2.To identify ML/DL model development challenges 3.To detection iterations and events that trigger these iterations	Case study	1. Interviews 2. Observation	1.Semi-structured interviews 2. Observation from workshops and meetings	Companies (3) Use cases (6)
2. What are the best practices, challenges experienced and architectural choices made by practitioners when deploying and operationalizing ML/DL models?	1.Development of a generic framework to deploy AI at the edge 2.Presentation of two architectural variants 3.Identification of key challenges when selecting an architectural alternative	Action Research	1. Internship	1. Semi-structured Interviews 2. Follow-up interviews 2. Observation from workshops and meetings	Companies (5) Cases (4)
	1.Identification of key factors for selection of specific architecture for AI deployment 2. Development of architecture selection framework	Case study	1. Interviews 2. Observation	1. Semi-structured interviews 2. Focus- group interviews 3. Workshops	Companies (7)
	1.Offer state-of-the-art and state-of-practice and experience of practitioners 2.Built a framework derived from literature for end-to-end deployment process	1. Systematic Literature Reviews 2. Grey Literature Reviews			
3. How can MLOps practices support large-scale embedded systems companies in achieving continuous delivery and evolve models?	1.Present state-of-art on adoption of MLOps in practice 2. Identification of different steps companies take when evolving their MLOps practices 3. Mapping of the companies to stages in the maturity model	1. Systematic Literature Reviews 2. Grey Literature Reviews 3.Case study	1. Interviews 2. Observation	1. Semi-structured interviews 2. Stand-up meetings 3. Workshop	Companies (3) Cases (3)
	1. Built conceptual framework for AI-driven business development framework 2. Identification of decision points for business case termination	Case study	1. Interviews 2. Observations	1.Semi-structured interviews 2. Observation from workshops, meetings and events	Companies (6) Cases (9)

on this, chapters 4, 5, 6, 7, 8 and 9 provide an overview of the findings obtained. Chapters 4 and 5 provide frameworks for designing the development process of ML /DL models and facilitating AI-driven business development. Chapter 6 presents architectural alternatives for deploying AI at the edge and Chapter 7 explains key factors that may influence these architectural choices. Chapter 8 seeks to improve understanding of the deployment and integration of ML/DL models, and Chapter 9 seeks to reduce the gap between ML and operations teams (MLOps).

4 Developing ML/DL Models: A Design Framework

This chapter has earlier been published as
M. M. John, H. H. Olsson, and J. Bosch, "Developing ML/DL Models: A Design Framework," In Proceedings of the International Conference on Software and System Processes, pp. 1–10, 2020.

Artificial intelligence (AI) has been popular in the recent years due to advancements in Machine Learning (ML) [100], Deep Learning (DL) [101], Big data and Cloud Computing technologies. ML/DL has emerged as a strong method of choice in many fields of science and technology, including robotics, speech recognition, natural language processing [102], computer vision [103] [104], etc. As a result, organizations are eager to integrate these innovations into software-intensive systems to increase value delivery. In larger organizations such as Google [105], Apple [106], Microsoft [18] and Facebook [107], ML/DL technologies have been widely used in services such as Google Translator, Google Street View, Siri, Bing search, Cortana virtual assistant, DeepFace, etc. Although ML/DL techniques are popular due to the sheer size of the available data and as they provide exciting data-driven decision-making approaches for useful insights, predictions and decisions, only very few people possess the knowledge needed to develop these solutions [21]. Typically, data scientists have expertise in areas such as mathematics, statistics, AI, ML, DL, Big data and Cloud computing. Due to a shortage of these skilled ML/DL experts - software developers, business analysts, data analysts etc., are often needed to take on the responsibilities normally performed by data scientists when data science teams are formed in organizations [28] [22]. In practice, software developers find it difficult to apply ML/DL techniques even when they see the value of their implementation. Moreover, non-experts are

building and deploying potentially invalid ML/DL models to meet their real-world needs by perceiving percentage accuracy as the only metric of model performance [21]. Alarmingly, non-experts are more satisfied with the learning outcomes and more confident in ML/DL models compared to the experts. Despite the level of expertise, it is clear that people in all domains are struggling to develop high-performance ML/DL models in the absence of a well-established design process.

With the advent of ML/DL technology, software organizations need to evolve their development practices and processes to incorporate intelligence in their products. Although there are many methods and processes available to support developers in traditional software development, this is not the case for ML/DL models. While the development, deployment and maintenance of the ML/DL system is recognized as a huge challenge in previous studies [10] [15], there is a need for systematic and structured approach to the implementation of ML/DL models. ML/DL is quite under-developed and requires further work to facilitate the development of high-quality systems compared with other fields, such as software engineering (SE) or database technology. Systematic and structured design process derived from the activities of expert data scientists in different organizations is required to make ML/DL development accessible to people beyond expert data scientists. The contribution of this paper is threefold. First, we identify and present the seven phases of the design process based on the various activities of expert data scientists in different organizations to support the development of well-performing ML/DL models. Second, we identify the key challenges that data scientists experience at each phase of the design process when developing the ML/DL model. Finally, we present the identified iterations between phases and the events that trigger these iterations to optimize the design process.

The rest of this paper is organized into five sections. In Section 2, we review related literature. Section 3 introduces the adopted research methodology and the case companies involved in our study. Section 4 presents the results of our case study. Finally, we conclude the paper and provide an outline of future research in section 5.

4.1 Related Works

4.1.1 ML/DL

Life has changed with the speed with which ML/DL technology has been implemented in real-world scenarios. Methods, recent advances and research opportunities in ML are described in [100] and an overview of DL is presented in [101]. DL differs from ML because it learns to automatically represent data using multiple abstraction layers [108]. Compared to DL, a large amount of work is expended on feature engineering to manually build this representation in ML. ML/DL is different from conventional SE because its behaviour depends heavily on external data. The main difference between ML/DL and non-ML/DL systems is that data replaces code within a ML/DL system and automatically identifies data patterns using a learning algorithm instead of hard-coding.

4.1.2 Developing ML/DL Models

ML/DL systems differ from traditional software in three ways. First, collecting, processing and updating data is time-consuming as models learn from data in ML/DL. Second, in addition to SE skills, teams need an in-depth knowledge of ML/DL technology to build high-performance models. Third, extra effort is required to deploy and place the model in operation compared to traditional software development. Three categories of people are involved in developing ML models [21]: experts, intermediate users and amateurs. Experts have deep knowledge of developing, deploying and operating ML/DL models. Intermediate users have less ML expertise and lack deep knowledge compared to experts. Amateurs are non-experts with no knowledge of ML/DL technology. Implementation of ML applications by software engineers is investigated in [109]. Very little research has been done on the work of intermediate users and amateurs. When automatic ML approaches fail or deliver unsatisfactory results [110], amateurs (domain experts and end-users) without ML expertise are actively involved in ML. The evidence indicates that even expert programmers experienced difficulty in applying ML [13]. Different kinds of tools are available to support the performance of people involved in ML/DL technologies. Although ML/DL offers powerful tools, the knowledge required to apply these tools to specific situations remains a challenge. Smaller non-profitable organizations have extensive local and domain knowledge available, but they

lack the skills to apply ML/DL to their problems [29]. Current working practices of data scientists and how AutoAI (Automated AI) influences these practices are described in [22]. Data processing approaches adopted by data scientists is proposed in [28]. According to [18], workflow used for ML/DL has nine stages which include data-oriented and model-oriented stages. A high-level description of the ML process workflow is described in [13]. ML workflow defined in [111] has four stages with associated activities namely, Data Management, Model Learning, Model Verification and Model Deployment. The scope of these workflows begins after the business case specification. These are similar to workflows defined in the context of data science, such as CRISP-DM (CRoss-Industry Standard Process) [58] and KDD (Knowledge Discovery Databases) [57]. CRISP-DM helps organizations to apply data mining in real-world scenarios independent of technology. Using data mining tasks, KDD is used to discover knowledge from data. The data centric essence and multiple feedback loops between different stages are part of all of these workflows.

4.1.3 Challenges in Developing ML/DL Models

As a consequence, people engaged with ML/DL model development encounter several challenges. Reference [10] outlines an attempt to identify and classify various software engineering challenges that exists when building and deploying ML components in software-intensive systems. Experts encounter problems when building systems using ML algorithms, as described in [13] [14]. The ML model development by non-experts is investigated in [21] based on their experience, knowledge and blind spots. The unique potentials and pitfalls of non-experts are revealed in this work. Various ML specific risks that should be taken into consideration during system design using SE framework of technical debt is illustrated in [15]. These include boundary erosion, hidden feedback loops, data dependencies, configuration debt, changes in external world, etc. Many studies have been conducted to test software [59] and ML models. However, studies considering the combination of both SE and ML have not been so intensively researched [60]. The implementation of a production-ready ML system needs testing, as does the software [19]. Challenges in the intersection of SE and ML has been described in [9]. They can be grouped into development, production, and organizational challenges. DL techniques need special infrastructure support compared to ML and traditional soft-

ware development [61] as scaling up DL models improve performance.

4.2 Research Method

Following the guidelines of Runeson and Höst [83], a multiple-case study was conducted to analyse the activities of experienced data scientists when designing ML/DL models in real-world settings. Case study is an empirical research methodology that focuses on the in-depth investigation of a contemporary phenomenon that is difficult to study separately in its real-life context [112]. In SE, case studies are conducted to understand, clarify or demonstrate new technique capabilities, method, tool, process, technology or organizational structure. Multiple-case study provides a deeper understanding of individual cases as a unit by comparing their similarities and differences [113]. In this paper, a multiple-case study approach was selected to study the similarities and differences in ML/DL model development practices followed by data scientists in different software-intensive embedded organizations.

4.2.1 Case Companies

In this section, we present three case companies and different ML/DL cases studied in each company as part of our study. All the reported cases are using real-time datasets.

Case Company A - Telecom: In case company A, we studied four cases in different business areas.
A1. Log Analysis: With the advent of ML/DL technologies, log data can be used to obtain useful insights. Logs generated by different products are retrained by reusing all existing infrastructure, models and parameters to generate profits for the organization. The data set consists of gigabytes of log files generated by different products within the organization.

A2. Paging Cell Phone: To access a specific cell phone, there is a need to page one or more base stations. The organization analyses the mobility patterns of user equipment (UE) within the network to page the cell phone when it is in idle mode more precisely using the existing paging algorithm. The data set includes all events that occurred for a specific node (approximately one TB of data) within one week in the organization.

A3. Detecting Garbled Speech Frame: When voice data is encoded over audio, speech frames are encoded. Certain flags in a specific speech frame contain information about the previous and subsequent frames. The organization predicts whether speech frame is garbled or not by analysing this information. The ML model should be super-fast and accurate to be placed in the system. The data set consists of speech frames collected from the Telecom industry.

A4. Predicting hardware faults: The organization predicts hardware faults with an intention to reduce the amount of hardware returned by the customer when it comes to repair. The information collected from crash log reports of different customers helps build the ML model. During hardware screening, they focus on two aspects : a) if the hardware is found to be faultless, it is returned to the customer b) if the the hardware has faults, it is sent to the repair centre. The data set consists of crash log reports collected from the telecom industry.

Case Company B - Automotive - *Autonomous driving vehicles*: The vehicle needs faster DL algorithms when driving on high-speed roads. In addition, the organization needs to ensure that the rate of failure for these safety critical products is minimal. The organization develops DL models in collaboration with partners. The data set consists of millions of manually generated test cases based on accident statistics or based on the fabrication of typical traffic situations; it is then mutated or combined in various ways. Furthermore, it contains thousands of hours of data recorded while driving in real life.

Case Company C - Packaging - *Defect Detection*: The organization utilizes DL models to detect defects in packages at each customer site during processing. Learnings from local training at each customer site is used to train the global model in the cloud. The global learning is then pushed back to the customer sites for inference through transfer learning to detect customer-specific defects more effectively. The data set consists of packages with different patterns and types.

4.2.2 Data Collection

The research we report builds on a semi-structured interview study conducted between September and December 2019 following an interview guideline. The interview guide consisted of three parts. Part I focused on the role and experience of the interviewee. Part II focused on current development practices and activities related to the development of ML/DL models. Finally, Part III explored the challenges experienced by data scientists in the design of ML/DL models. Our interview study involved experts from three large embedded systems companies that represent different business areas and domains. All interviews lasted one hour and were conducted via video conferencing. All interviews were recorded and transcribed for analysis with the consent of the interviewees. In addition to the interviews, we organized a total of five workshops (representatives from different companies participated in a number of workshops together) at the case companies, where we presented our research and had the opportunity to get feedback from the company representatives who also shared their experience, discussed the research scope and contributed their knowledge on ML/DL model development. Furthermore, the workshops provided good opportunities to explore many of the challenges involved in the development of ML/DL models in detail. In addition to interviews and workshops, we also conducted several meetings with representatives at the case companies to improve the quality of our research study. The experts involved in study are summarized in Table I where *I** denotes Interview participants and *W** denotes Workshop participants.

4.2.3 Data Analysis

The interview summary was cross-checked with audio recordings, interview transcripts and notes taken during the interviews and workshops. We applied elements of open coding to analyse the interview summary in order to identify key concepts and subsequently group them into conceptual categories [97]. We used triangulation to enhance the precision and to strengthen the validity of our research study [98], which primarily relies on qualitative data. As we were three authors conducting the interviews and reading the transcripts, triangulation was used to take multiple perspectives on the subject under study; thus it provides a broader picture [99]. The objective of the first iteration was to identify the different phases involved in the ML/DL model development. The second iteration was performed

Table 3: Description of Experts involved in interviews and workshops

Case company	Experts	
	ID	***Roles***
A	I1	Senior Data Scientist
	I2	Senior Data Scientist
	I3	Instigator
	I4	Data Scientist
	W1	Data Scientist
B	I5	Technology Specialist AI/ML
	W2	Director
	W3	Manager
	W4	Software System Architect
	W5	Research Engineer
C	I6	Data Science Manager
	I7	Solution Architect
	I8	Data Scientist Analyst
	I9	AI Application Specialist
	W6	Head of Data Science
	W7	Director
	W8	Manager
	W9	Technology Specialist

to indicate iterations and triggers between phases to optimize the design process. The third iteration was carried out to identify the challenges in developing ML/DL models with the cases involved in our study. The results deduced from the analysis were sent to the experts involved in the study for validation.

4.2.4 Threats to Validity

We discuss our efforts in mitigating validity threats in particular to construct validity, reliability and external validity related to our study. To improve construct validity of our research, the authors and participants are most familiar with development of ML/DL models. We used multiple techniques (interviews, workshops, meetings) and multiple sources (data scientist, technology specialist, AI application specialist, etc.) for data collection. Our research was on three case companies, with the results reviewed by practitioners directly and indirectly involved in our study. To

ensure reliability, the findings were validated with other experts at the company. Regarding external validity, we believe that the main contribution of this work is applicable to all companies that are interested in integrating ML/DL models into their software-intensive embedded organizations.

4.3 Case Findings and Analysis

Based on our interview findings, workshops and meetings with the three case companies involved in our study, we identify seven typical phases that data scientists undergo when developing ML/DL models: (i) Business case Specification, (ii) Data Exploration, (iii) Feature Engineering, (iv) Experimentation, (v) Development, (vi) Deployment and (vii) operational. These phases are used to develop a systematic and structured framework for the design process of ML/DL models in software-intensive embedded organizations. Fig. 29. shows an overview of the design process, while Table II presents a summary of the challenges associated with each phase. In the sections below, we detail each phase.

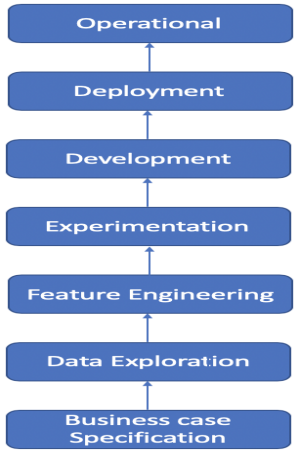

Figure 3: The phases involved in designing ML/DL models.

4.3.1 Business Case Specification

Purpose: Specifying the business case for ML/DL projects is challenging since costs and benefits are more difficult to predict compared to traditional development.
Approaches: When profitable business cases are proposed, we observe that

data scientists spend one or two days running simple ML/DL models over a structured dataset to check whether or not business case criteria can be met. They schedule discussions with product owners to see whether the accuracy obtained is worth it or to spend time improving accuracy. Data scientists also verify that sufficient data is available and accessible for case implementation. Large organizations make use of their own dataset for implementing business cases.

> I3: If we think it is a business case, then we typically think do we have the data to even use ML/DL for this project.

Challenges: 1) **High Costs Involved**: Implementation of ML/DL models involves infrastructure costs, the need for data scientists and domain experts as compared to traditional development. For instance, building an infrastructure for collecting and examining radio data for the implementation of an analysis tool involves high costs. Unfortunately, certain business cases can only be solved with ML/DL, even though they involve high costs.

> I3: It is very hard to nail down exactly the business case when it comes to ML/DL because, typically are a lot of costs involved. There are more costs in ML/DL projects than ordinary projects.

2) **Communication Gap with Stakeholders**: Communication with stakeholders needs to be strengthened to improve value delivery. In our study, we note that it is difficult for data scientists to understand the needs of stakeholders.

> I4: We are assigned the business case, but we never get any information about the quantitative specifications to be optimized.....In order to make good business value, we need input from stakeholders, and sometimes we need to play an educational role because they are not aware of the potential risks involved in the design ML/DL systems.

3) **High Expectations of AI**: Organizations have high expectations of AI, so it is feasible to create an early insight into what can be achieved with a specific business case. For instance, when product owners hear the word AI, they have super high expectations. So, it is a good idea to build an intuition on the attainable theoretical maximum limit in terms of the accuracy

for the business case.
4) **Less Data Scientists Availability**: As data scientists are scarce resources, so they should be assigned to projects of high business value. For instance, we find that the number of data scientists is much lower than the number of other technical staff in case companies involved in our study. Therefore, proper utilization of the available data scientists is important.
5) **Large Dataset Needed**: For better results, DL models need a large dataset compared to ML models. For instance, datasets of safety-critical products are collected by manually building, mutating and simulating test cases, and recording thousands of hours of data while driving on roads to show that failure rates are low.

> I5: Need enormous amount of data for perception systems, so it is hard to collect all that data.

4.3.2 Data Exploration

Purpose: Analysis of the dataset to identify characteristics, patterns and points of interest.
Approaches: We note that data scientists spend time with domain experts to understand both the data and the system. For instance, *I4* reported that data scientists schedule frequent meetings with domain experts to get suggestions on potential data aspects to be considered in the dataset. They either use domain knowledge experts or rely on unsupervised or clustering techniques to label the dataset. We notice that stakeholders provide datasets for exploration in some organizations. For instance, partners of the organization provide infrastructure to access the data for perception.

Data scientists continuously build and test hypotheses around the data for a better understanding of data and system. It also provides insight into non-functional requirements, such as computational budgets, model expectations, etc.

> I2: We always start with data, look at the data and just explore the data as much as physically possible. Just to understand it and and to get some intuition around what is actually going on and what is behind all this data .... It is highly unstructured in the way that we really do not know where the data will take us, so we try building and experimenting with the hypothesis that we have around data.

We find that data scientists select a small sample from a large dataset and try to remove outliers. They then use visualization techniques to understand the data distributions and slowly scale the dataset so that, finally, it will result in a dataset that is completely understandable.
Challenges: 1) **Privacy concerns and Noisy data**: The availability and accessibility of data is limited due to privacy concerns and noisy data. As a result, data scientists are forced to spend more time trying to understand data and the system. For instance, as access to node configuration is restricted and data is anonymized, data scientists spend three months analysing dataset events to reverse engineer the baseline.
2) **Shortage in Domain Experts Availability**: Consulting domain experts is difficult for data scientists in larger organization as their numbers are fewer. *I1* proposed that it would be a good idea for domain expert or data scientists with domain expertise to sit with the team to gain an understanding of the data and the system. That is impossible in organizations due to a lack of proper allocation of domain experts to ML/DL projects.
3) **Labeling**: The dataset available to data scientists is either partially or completely unlabeled all the time. Since the availability of domain experts for labeling is a challenge, some organizations, for instance, rely on interns, especially students, to add labels. Consequently, confidence in labeling decreases.

> I3:Very, very often you want label data, and that is frequently scarce

> I1:Even then, as a data scientist, you have to use the domain knowledge expert to label it

4.3.3 Feature Engineering

Purpose: The identification of relevant features to ensure that the ML/DL model has accurate information for inference.
Approaches: Most data scientists consider feature engineering as an important phase and compose features with the help of domain experts.

> I3:Identifying and modelling the right features is quite relevant and has quite a bit of impact as well

A typical approach followed by data scientists is to start with a high-dimensional feature set and scale down to a few-dimensional set by elimi-

nating irrelevant features that do not affect model performance. In contrast to this, data scientists in some organizations scale from a low-dimensional to a high-dimensional feature set by adding relevant features. We noticed that data scientists add features that are not directly part of the dataset to the feature set based on domain knowledge perspective. For instance, to predict the presence of a UE in the network, the notion of time is added to the feature set when they realize that it is a reasonable property that people will not move a lot at night.

> I1: We do this feature engineering both as part of before doing experimentation and as an integral part of the process

Challenges: 1) **Increasing Complexity**: Increasing the complexity of features in ML/DL models entails costs. Therefore, a careful evaluation is required to select the best features in a reasonable amount of time.

> I4: Adding a new feature is costly in the way we work. So, we want to have an idea of what this feature can bring us before we do the implementation

2) **Improper Feature Selection**: A feature set including irrelevant features affects the model performance. Without a proper understanding of each feature, data scientists find it hard to add relevant features to the feature set. For instance, it is hard to select the relevant feature if two features contain the same information.

4.3.4 Experimentation

Purpose: To compare the performance of multiple algorithms to find a specific or specific set of ML/DL models that best fits the business case.
Approaches: We see that data scientists follow certain guidelines for ML/DL model implementation. They use classical ML techniques for smaller datasets, DL for large labeled datasets, and clustering techniques when clusters are clear during data exploration. They apply random forest to small dimensional problems and pairwise correlation between features for relatively small dimension problems.

> I1:There is no scientific precise method for that; it is completely based on one problem to another problem. That is not well-defined

Data scientists rely on a state-of-the-art approach to find the current best algorithms for research-oriented projects and well-known and explored algorithms for understanding the process of small-scale pilot products. For instance, as object classification is a key problem for autonomous driving vehicles, data scientists are looking for the state-of-the-art learning and are selecting the best available algorithm. We also observe that the case companies follow different approaches to state-of-the-art learning. Most companies rely on published papers, while a few data scientists in some companies attend prestigious conferences in their field of work and then share any information gained from these conferences with other data scientists through study sessions.

Typically, data scientists take a small sample from a large data set and identify the class of problem as if it is an image/text etc. Then they pick two to three approaches, ranging from simple and classical to more advanced approaches, such as DL, that have traditionally performed well in the identified problem class. For instance, Näive Bayesian models are used for text classification as they worked well for spam filtering, sentimental analysis, etc. The various approaches are experimented and validated, and subsequently compared to different dimensions. Data scientists also adopt a hypothesis-driven approach to introduce a specific feature to the model that can iteratively enhance the performance. We also noted that data scientists are interested in automating the experimentation tasks using tools such as H2O.ai, AutoML, Auto-WEKA etc.

> I2: Instead of sitting down and trying manually, I would be looking into some kind of tool that could automate this for me

Challenges: 1) **Introduce Bias**: Data scientists are introducing bias based on their experience in algorithm selection. For instance, most data scientists tend to work with Näive Bayesian Models, Logistic Regression, Tree Models, Random Forest, Support Vector Machines etc. as favourite models to obtain base performance. We observed a conflict in the study, as one of the interviewees referred to the random forest as being computationally fast to train and to obtain results in seconds, while another interviewee indicated that the random forest is flexible but less interpretable.

> I1: They bring their own experiences and then introduce own biases into the whole working

2) **Uncertainty in Selecting Algorithms**: Data scientists are quite uncertain about how to select the best algorithms for a specific business case. For instance, they often try very expensive trial and error methods for months, which results in huge wastage of time and resources.

> I1: If you have labels on your data, and if the question is should you always use supervised learning or not, I would never say yes or no to that

When choosing algorithms for perception problems that require real-time performance, data scientists should be extremely careful. For instance, in autonomous driving vehicles, the chosen algorithms should be fast enough to be executable in real-time.
3) **High Complexity of DL Models**: Although DL models are popular, most data scientists prefer ML models as they are less complex.

> I3: There is too much black magic in that DL models. Like, it feels to me that it depends more on luck than actually thinking about the problem. So, I am not very fond of these models

> I5: We need to have a lot of AI and DL knowledge to understand the limitations of the networks, when they may fail a lot ..... We are only half done after we have trained the networks

4) **Need for Deep DL Knowledge**: Data scientists need deep DL knowledge compared to ML models. For instance, regarding perception, deep DL knowledge is required to understand the network limitation, network merging, activation function, etc.

> I3: For me, it is harder to understand what the individual layers in a longer DL pipeline does

5) **Related Work**: We note that in some organizations, interns are asked to look into state-of-the-art approach or a particular concept because data scientists are busy with other tasks. Some data scientists consider the published work as toy examples. For instance, the related work in log anomaly detection did not take into account the diversity of logs and log styles.

4.3.5 Development

Purpose: Tuning of selected algorithm/set of algorithms to the parameters. *Approaches:* Data scientists adopt hyperparameter tuning to optimize the model. These hyperparameters can be based on previous literature or any findings or experiments that data scientists would like to make. Most models tend to have a high-performance hyperparameter set. For instance, the DL model has a good hyperparameter set for a gradient optimizer like Adam, which has historically performed well for image recognition.

Data scientists believe that the final model selection depends on the end goal requirements. Requirements can be accuracy, prediction time, available computational resources, understandability, interpretability, verification and validation, interoperability with other environments, reproducibility and the model execution environment. For instance, choose an algorithm that suits the hardware performance available during the lifespan of a specific product development in a perception system. In the case of garbled speech frames, accuracy is given greater importance since misclassifying the correct frame is much worse than not classifying the wrong frame. In a different scenario, ML analyses customer trouble reports in order to determine the team to which the report should be forwarded. It will take a while before an individual examines the report when it is generated. In this case, prediction time is not really important.

We also observe that data scientists have different attitudes when interacting with ML/DL models. Some data scientists do not care how the model works as long as it works, while others use accessible domain knowledge on simple models, which leads to an easily explainable model.

> I1: Understanding a specific model or how we have trained the specific model is of great important to us

Challenges:1) **Determining Final Model**: Data scientists have experienced some confusion when deciding the final model that will be ready for production based on end goal requirements.

> I2: It is really iterative in the sense that even in the final stage we think we have a good solution, it could still end up that we realise that we need to go back to data side and redo things or change the architecture

> I5: It costs a lot of money to change the algorithm in such a large project because a lot of validation is going on in the case of safety critical products

2) **Model Execution Environment**: The training phase is restricted to the model execution environment in software-intensive embedded systems. For instance, paging is carried out at a high volume with strict latency requirements.
3) **More hyperparameter Settings**: When dealing with DL models, more hyperparameters need to be set for better results. For instance, hyperparameters such as learning rate, number of neurons per layer, activation function, etc., should be set for DL models.
4) **Verification and Validation**: V&V activities are significant in the design of ML/DL models. For instance, there should be greater degrees of certainty that the system performs well in the case of safety-critical products, since it is likely to injure or kill people.

> I5: We need to prove that the failure rates are low enough

4.3.6 Deployment

Purpose: Integrating the ML/DL model into the production environment to make realistic data-based decisions.
Approaches: We see that data scientists place some requirements on the ML/DL models prior to deployment. They ensure code review, unit testing of all components, ready-to-use production infrastructure, proper review of model training, check if model build by a data scientist is understandable to another data scientist or not, maintenance, robustness, and stability test.

> I4: We want to be a team that is not only like a data scientist team that train models, but we also want to be experts on how to deploy models

Data scientists focus on three things to bring the model into production. First, prepare code ready to be put in the docker container. Second, plan for integration with internal systems, and, third, use the reusable functionality existing in API services to wrap up the model.

> I1: Build with production in mind from scratch

> I3: When that is in production, it is not that big of a difference towards ordinary software components, but getting it into place is quite painful

We found that data scientists place the model under close supervision of A/B testing. Organizations also use a model execution environment with built-in support for A/B testing, canary selection etc., as an in-depth part of the product. The services deployed by data scientists have a training interface, inference and evaluation interface to train the data, kick-off retraining with new data and compare new model and old model to choose whether or not to roll back. Ideally, *I2* would like to have a mechanism to define proxy value to be evaluated to check whether the model is worth deploying or is worth redeploying or retraining. Often, data scientists adopt an approach where the old model has been adapted with new functionality, resulting in a newer version of the model placed in the same infrastructure. Then a comparison between the old and new model takes place to choose the best model. In contrast to this approach, data scientists perform continuous retraining by adding functionality to the existing model. Data scientists introduce model decay to automatically update the model. For instance, as UE moves from base station to base station, it increases the value between them. Divide the increasing probability of every transition by half a week and remove the base station at zero.

Challenges: 1) **Less DL Deployment**: In comparison to DL models, data scientists prefer less complex ML models because they can be quickly deployed.

> I2: I never actually needed to deploy a DL system or a neural network at all, in general. There are many other, much more efficient algorithms than neural networks.....for production systems, it's 100% traditional ML

2) **Integration issues**: Integrating ML/DL components into a software-intensive system is challenging, as ML/DL model is a small part of the entire system. For instance, it is difficult to integrate individual DL models in a perception system.
3) **Internal Deployment**: Since the customer network and the organization have a strong boundary, the deployment takes place within the organization rather than within the customer networks. For instance, it is impossible to collect and modify data in customer networks after the product has

been purchased. Although desirable, it proves challenging to allow small deployments on small customer subsets.
4) **Need for an Intelligible Model**: There is a need to ensure that other data scientists understand the model developed by a particular data scientist.

> I3: It is very hard for another data scientist to understand the model in the case of some models that we have taken over from a data scientist

5) **Model drifts**: Model drifts should be properly identified to decide when to retrain or update. For instance, models are retrained on a daily, weekly, monthly or year-on-year basis based on the domain.

> I1: Putting it into production and having it up there 24/7 is a completely different ball game

4.3.7 Operational

Purpose: Once the model is deployed, it needs to be monitored for performance in the field.
Approaches: Data scientists perform continuous monitoring and logging activities to make sure the model performs as expected. They also ensure that the deployed ML/DL model in operational is well integrated with other software components as well as the data pipelines set up in the software-intensive embedded organizations.

> I1: Most people focus on the algorithm and modelling part and lose out on operational aspects

> I3: May be what is a bit specific putting ML/DL models in production is that you want some way to actually monitor that the model keeps performing the way you need it to

Challenges: 1) **Training-Serving Skew**: Training-serving skew is a disparity between training and serving performance. As an instance, the system is trained on events to ensure that it performs well in safety-critical products and is less suited to real-world situations during serving.

> I5: It is very hard to capture real data for these long-tailored events because some of them will happen so rarely.

2) **Communication with End-users**: The need to encourage further communication with end-users is emphasised by data scientists as it is hard for them to understand the model. For instance, for the analysis tool designed to analyse radio data, data scientists have answered all the questions raised by customers because they do not have strong confidence in AI. In some cases, data scientists work with testers who have no experience in developing software. As a result, these testers find it hard to understand the models.

> I4: We need to educate them, because they are very interested in understanding why we have done some specific predictions

> I3: If you take an ordinary software component no one really questions it as it is conceptually quite understandable; but with ML/DL models, there are many more questions.

3) **Maintain robustness**: Maintaining robustness in ML/DL models is complex compared to traditional software because ML/DL models are more data-intensive. For instance, it is difficult to maintain the robustness of data pipelines - especially in distributed systems as Telecom.

4.3.8 Iterations and triggers

The design process derived from the activities of experienced data scientists in different organizations can act as a standard design framework when developing ML/DL models. Based on our interviews and workshops, we identified several iterations between phases of the design process, as shown in Fig. 6. Although following the design process helps with developing efficient ML/DL models, leveraging the iterations can help in building even better models by ensuring better prediction, efficient inference and high business value. Iterations optimize the already proposed design process. Below, we outline these iterations and the events that trigger those iterations.

1) *Deployment to Development phase*: The first iteration is from deployment to development phase. We have noticed that deploying the model to production is not a one-time activity in most software-intensive embedded organizations. Rather, it is a continuous process. The model makes accurate predictions if the data used for prediction and training have a similar data distribution. To mitigate data drifts or to keep the model

Table 4: Challenges associated with each phase

Phases	Challenges
Business Case Specification	High Costs
	Communication Gap
	High AI Expectations
	Less Data Scientists
	Large Dataset Needed
Data Exploration	Privacy concerns and Noisy data
	Shortage in Domain experts
	Labeling
Feature Engineering	Increasing Complexity
	Improper Feature Selection
Experimentation	Introduce Bias
	Uncertainty in Algorithm Selection
	High Complex DL Models
	Need for Deep DL Knowledge
	Related Work
Development	Determining Final Model
	Model Execution Environment
	More hyperparameter Settings
	Verification and Validation
Deployment	Less DL Deployment
	Integration Issues
	Internal Deployment
	Need for an Intelligible Model
operational	Training-Serving Skew
	End-user Communication
	Model Drifts
	Maintain Robustness

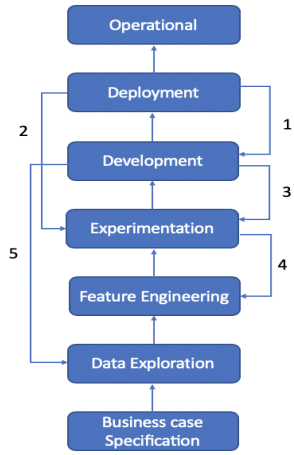

Figure 4: Iterations to Optimize the Design process

up to date, this iteration is triggered. The simplest solution is to retrain the model with new data and validate the model to ensure that the model still provides accurate results. The learning algorithm and hyper parameter space remain the same as the trained model when retraining with new data.

2) *Deployment to Experimentation phase*: The second iteration is from deployment to experimentation phase. Once the model is put into production, it makes real data predictions. We find that data scientists are looking at a state-of-the-art approach to find better learning algorithms as a replacement for the well-performing deployed model. Replacement takes place to optimize prediction accuracy, hardware cost, explainability, latency, prediction time, etc. This iteration will only be triggered if data scientists confirm that new replacement algorithms perform much better than the deployed model and increase business value. If so, the model must be trained with a new learning algorithm on already existing data and features and finally proceed with the redeployment of the model.

3) *Development to Experimentation phase*: The third iteration is from development to experimentation phase. Since a lot of ML/DL algorithms are available today, data scientists are still experimenting with different algorithms in search of an optimal algorithm that is appropriate for their business case, even if they have finalized a well-performing algorithm. This iteration is triggered, when it is assumed that experimenting with an-

other algorithm before proceeding to deployment will improve the model performance even if the algorithm is finalized. In this iteration, the model is built and trained using the same data and features, but with a different algorithm and hyperparameter tuning. Once the model has been finalized, it will continue to be deployed.

4) *Experimentation to Feature Engineering phase*: The fourth iteration is from experimentation to feature engineering phase. While experimenting with different algorithms, data scientists understand the relevance of features and the correlation between features. As a result, this iteration is triggered when it is assumed that the model performance can be better improved by adding or removing specific features to/from the model that are directly or indirectly dependent on the dataset. Iteration begins with the extraction of features, model learning and continues to the deployment and production phase, if better results are achieved.

5) *Development to Data Exploration phase*: The fifth iteration is from development to data exploration phase. Even after a model is finalized, this iteration is triggered when it is assumed that there are opportunities to improve model performance by re-examining some of the data aspects. If some interesting aspects are coined after exploring data sets, then extract the features from the same dataset, build and train the model from scratch and proceed to deployment.

4.4 Conclusion

Although ML/DL technologies are gaining an increasing interest in industry, organizations face several challenges when developing these models. In this paper, we identify seven different phases that data scientists move through when developing ML/DL models, and we detail the challenges they face in each of these phases. In addition, the study identifies the iterations that occur in between different phases and the events that trigger these iterations to optimize the design process when developing models. We believe that understanding the systematic and structured design process derived from the activities of expert data scientists in different organizations can make ML/DL model development accessible to people beyond formally trained data scientists. In future research, we plan to expand our interview study by involving additional case companies with more experts and further validate our findings with experts in the field.

5 Towards AI-Driven Business Development Framework

This chapter has earlier been accepted as
M. M. John, H. H. Olsson, and J. Bosch, "Towards an AI-driven Business Development Framework," In ICSSP Special issue of the Journal of Software: Evolution and Process, 2021.

Digitalization signifies a transition from hardware and product-based business to one that relies primarily on software, data and Artificial Intelligence (AI) to boost products, offer pure software-based products and provide customers with new digital and data-driven services [40]. In addition, and with the help of digital technologies, companies can significantly accelerate value creation, replace transactional business models with more service-oriented business models and shift towards a continuous customer relationship characterized by completely new ways of retrieving, responding to and redefining customer and market needs [114]. Digital and data-driven services are used in several domains, including preventive maintenance of vehicles, mobility services focused on subscription and "pay-as-you-go" business models, automation and as a key component in autonomous driving, etc. [114]. However, during the transition from hardware and product-based business to a business that increasingly focuses on digital technology such as software, data and AI, companies need to evolve and enhance their current systems with new technologies. From this aspect, AI/ML/DL (Artificial Intelligence/Machine Learning/Deep Learning) technologies provide excellent opportunities as they allow for innovations and new ways to serve customer needs [114]. Moreover, these technologies allow companies, particularly in the field of embedded systems, to adapt to new ways of working characterized by continuous inte-

gration practices and the deployment of not only software features but also of data and ML/DL components [114].

Companies across domains are using AI/ML/DL [100] [101] technologies to improve, scale and optimize their businesses. Introducing AI/ML/DL in companies enhance their business scalability, productivity, customer intimacy and operational efficiencies. ML/DL technologies are effective because of the massive size of data available and because they provide opportunities for data-driven decision-making, business insights, pattern recognition and predictions, only a few people possess the skills required to develop these solutions [21]. Due to a shortage of data scientists, software developers and other roles and functions inside an organization are often asked to shoulder the responsibility of developing, deploying and evolving models [28] [22]. Even though software developers recognize the value of their implementation, they find it difficult to apply ML/DL techniques due to lack of expertise in data science [109]. Moreover, by seeing accuracy as the only evaluation metric of model performance, non-experts with little to no data science background run the risk of developing and deploying invalid ML/DL models for their day-to-day needs [21]. Regardless of the level of expertise and/or access to data scientists, companies across domains are struggling to create high-performing models in the absence of well-established and systematic design methods and processes. Software companies must evolve their development practices and processes to incorporate intelligence in their products as ML/DL technology matures. In conventional software development, there are several approaches and procedures to assist developers; however, this is not the case for ML/DL models. While previous research has identified the development, deployment, and evolution of ML/DL models, as well as their incorporation into larger systems as challenging [10] [15], there is little, if any, support on how to develop, deploy and continuously evolve ML/DL models in a systematic and structured way. In previous literature studies [54] [13] [115] [56] [57] [58], existing processes and structures that portray ML/DL development focus more on the the data-intensive context or on a mix of data and model requirements context and has little to no emphasise on business case generation, selection and validation.

In our previous research, and based on multi-case study research in three embedded systems companies "Developing ML/DL Models: A Design Framework" which was presented at the International Conference on Software and Systems Process (ICSSP) 2020 [116], we identified the typ-

ical phases that data scientists go through when designing ML/DL models. The seven typical phases are i) Business case Specification, (ii) Data Exploration, (iii) Feature Engineering, (iv) Experimentation, (v) Development, (vi) Deployment and (vii) Operational. For each phase, we identified the approaches as well as the key challenges that the data scientists involved in our study experienced. Finally, in order to optimize the design process, we outlined the iterations that occur between the various development phases, as well as the events that trigger these iterations.

This journal is an extension of the paper "Developing ML/DL Models: A Design Framework" (ICSSP) 2020 [116]. In this extension, we provide the following contributions in addition to the contributions already presented in the conference paper. First, in addition to the original three case companies, we add three new case companies to allow for additional empirical results and further generalization of our findings on developing, deploying and evolving ML/DL models. Second, we present a conceptual framework in which we outline not only the activities involved in the development of ML/DL models but also the roles and organizational functions involved, as well as the iterations that take place and the activities that trigger these to optimize the process. In this framework, the original phases identified in ICSSP are incorporated into three high-level activities (i.e. focused on business, data and models) that companies perform in parallel to develop, experiment with and optimize the models they develop. We offer a blueprint for how to effectively incorporate AI/ML/DL into the business of companies in a systematic way with this framework. The framework details how ML/DL business cases are generated, selected and validated by customers and practitioners, and how datasets needed for business cases are collected and explored to find interesting findings as well as how to coin important features, develop, deploy and evolve models. Finally, we show how our framework can assist in resolving the key challenges identified during our empirical study as well as exploring different decision points for immediate termination of business cases that do not provide significant cost-cutting, time-saving or any other benefits to customers. The framework depicts continuous delivery of ML/DL systems to accelerate AI-driven business with companies and offers an agile way of working rather than a sequential way of working to better approach the development, deployment and evolution. With AI-driven, we refer to the inclusion of ML/DL models in software-intensive systems with the intention of generating better results than with other (algorithmic) approaches.

The ML/DL model does better as it learns from data and in this way companies can accelerate results. The use of AI technologies helps companies improve their products and, in the end, their business. This can happen only if they manage to successfully incorporate the ML/DL development in the larger systems development and business context.

The rest of the paper is organized as follows: In Section 2, we review contemporary literature on how digital technologies are transforming the embedded systems domain and how AI/ML/DL can contribute to this transformation and its applications. In Section 3, we present the research method as well as the six case companies involved in the study. In Section 4, we report on our empirical findings from the multi-case study research in the six case companies, as well as the challenges they experienced. In Section 5, and based on our empirical findings, we derive a framework to help companies effectively incorporating ML/DL development into their business and the larger system of which the ML/DL component is a part. We review related work in Section 6 and discuss threats to validity in Section 7. In Section 8, we conclude the paper and provide an outline for future research.

5.1 Background

In this section, we first review contemporary literature on how digital technologies are transforming current business practices in embedded system domains, particularly with the introduction of DevOps, DataOps and MLOps practices. Second, we discuss previous studies on AI/ML/DL technologies and their applications in various industry sectors.

5.1.1 Digitalization: New technologies and Ways of working

Most companies have experienced rapid changes as they have become more digitalized in recent years [31]. According to Gartner [117], digitalization generates new revenue and value by utilizing digital technologies to change a business model. With sophisticated mechanisms to support massive data collection, processing and execution as well as novel ways to connect and communicate, digital technologies shape the way companies in the embedded systems domain operate and experience the emergence of new and faster business opportunities than ever before. The introduction of functionality in software rather than hardware enables companies to

enhance customer experience by upgrading and refining products as well as by extending product lifespan.

During recent years, and as reported in previous studies, continuous integration (CI) and continuous deployment (CD) practices lead to shorter development times and faster placement of release candidates in production environments. Much more recently, new ways of working have begun to emerge and gain traction for companies to leverage on what these new technologies aim to deliver. Practices such as DevOps [118] [119] [120], DataOps [43] and MLOps [121] [44] are being introduced with the intention of advancing product automation as well as quality in terms of development, data and ML operations. Penners and Dyck [122] propose DevOps as a collaboration of teams working in development and IT operations within a software-intensive organization to deliver faster software changes [118]. The main benefits of introducing DevOps in companies [119] include the improved delivery speed of software changes, increased productivity in operations work, higher quality, and better organizational culture and mindset. Companies, on the other hand, face several difficulties when adopting DevOps. They are inadequacies in infrastructure automation, high demand for skills and knowledge, project and resource constraints and monitoring challenges. According to reference [43], 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 chooses the best practices, processes, tools and technologies from Agile software engineering and DevOps to regulate analytics development, optimizing code verification, building and delivering new analytics and thereby fostering a culture of collaboration and continuous improvement". Some of the challenges encountered when introducing DataOps include a lack of pipeline robustness, data silos, organizational restructuring, etc. MLOps adopts and applies DevOps principles to ML models rather than software, bringing together the developmental cycles followed by data scientists and ML engineers with those of the operational teams to ensure consistent delivery of high-performance ML models [121]. One of the major challenges is educating for AI operations because most data scientists are not exactly computer scientists by education and most data-intensive companies have little to no idea on how to manage their data [123].

Digitalization is indeed much more than DevOps, DataOps and MLOps practices, but they do enable embedded systems companies to adopt shorter and continuous cycles with regards to software, data and ML technologies.

These are referred to as continuous development and deployment practices (for software, for data and ML models). This can also benefit the mechanics and electronics parts of the system (by for instance allowing for software updates to an existing system, providing means for effective use of data collected by the system and for e.g. developing predictive maintenance services etc.), as they are considered critical. Also, the case companies we work with within this study view these as core concepts and competences in order to accelerate digitalization and therefore, we view them important as mechanisms to help thrive in a digital world. The companies regard it as critical to integrate DevOps, DataOps and MLOps practices into their workflow because data and software components help bring additional value to their businesses by opening up new opportunities.

5.1.2 AI/ML/DL and its Applications

Due to advancements in ML, DL, Big data and Cloud computing, AI have grown in popularity. Jordan et al., describe [100] methods, recent advances and research opportunities in ML, while Goodfellow et al., [101] present an overview of DL concepts and techniques. ML analyzes and recognizes data patterns for value generation. DL, on the other hand, use multiple neural layers to learn from data. DL differs from ML in how it learns to represent data automatically using multiple abstraction layers [108]. In contrast to DL, ML necessitates a lot of feature engineering task to manually build this representation. ML/DL differs from conventional SE (software engineering) in that its behaviour is heavily reliant on external data. The primary distinction between ML/DL and non-ML/DL systems is that in ML/DL systems, data replace code and detect data patterns with an algorithm rather than hard-coding.

ML/DL has emerged as a preferred method in many fields, for instance, robotics, speech and image recognition, NLP (Natural Language Processing) [102], computer vision [103] [104], etc. Significantly, ML/DL technologies have been widely used to increase customer satisfaction in both large and small companies as well as online companies. For instance, ML/DL techniques have been widely by companies such as Google [105], Apple [106], Microsoft [18], Facebook, etc. [107] in their services such as Google Translator, Google Street View, Siri, Bing search, Cortana virtual assistant, DeepFace, and so on. Furthermore, ML/DL technologies are being adopted in a wide range of companies [124]. ML/DL applications

in the retail sector include recommendation engines, market segmentation, inventory planning, etc. In contrast, ML/DL is used in demand forecasting, condition monitoring, process optimization, etc. Some of the use cases in the health care sector are real-time warnings and diagnostics, disease and risk identification, etc. whereas in financial services, the use cases include evaluation of credit worthiness, risk analytics and regulation, customer segmentation, etc. ML/DL technologies are used in use cases for the energy and utility sector, such as power use analytics, carbon emission and trading, customer-specific pricing, etc. Adoption of ML/DL technologies is widespread and advanced in the online domain, for instance, King, Peltarion, etc. Furthermore, ML/DL is becoming more important for software-intensive embedded systems, for instance, image recognition and prediction services such as predictive maintenance [125], road planning, etc. Developing, deploying and evolving a complex ML-based business system continues to be a significant challenge for software-intensive embedded systems [15] [126].

5.2 Research Method

The research presented in this paper is part of a larger research initiative in which fifteen embedded systems domain companies and five Swedish universities collaborate to improve the digitalization capabilities within these companies[1]. In this context, we have conducted longitudinal multi-case study research with several of the companies on topics related to the development, deployment and evolution of AI/ML/DL technologies. For the purpose of this paper, we worked in close collaboration with five of these fifteen companies as well as with a company external to this collaboration, but was found to be highly relevant because it provides a platform for AI deployment and has several embedded systems companies as customers.

In our previous research, we conducted multiple case study research in three of the embedded systems companies (case company A, B and C) to explore and identify the key phases and challenges that data scientists face while developing, deploying and maintaining models [116]. As an extension to this previous research, and to advance the empirical insights as well as the conceptualization and generalization of results, we have added three additional case companies. Two of the three additional case

[1]Software Center - https://www.software-center.se/

companies, (case company D and E) are embedded system companies and one of them (case company F) is a non-embedded system company that is not part of the research initiative. We included case company F, in our study because they offer an end-to-end ML/DL deployment platform to e.g. embedded systems companies. Following the guidelines of Runeson and Höst [83], we conducted a multiple-case study to explore the activities involved in the development, deployment and evolution of ML/DL models. In particular, we were interested in the challenges that practitioners experience and how these can be mitigated by enhancing how ML/DL model development is incorporated into the overall business context of a company. The case study method is an empirical research approach that relies on an in-depth analysis of a contemporary phenomenon that is impossible to study independently in its real-life context [112]. Case studies are conducted in SE to understand, clarify or demonstrate new technique capabilities, methods, tools, processes or company structures. Multiple-case studies allow for an even deeper understanding of individual cases as a whole by comparing both their similarities and differences [113]. Throughout our study, we worked in close collaboration with practitioners from six different software-intensive embedded system companies to understand the activities they perform, the challenges they deal and the ways in which ML/DL model development is becoming increasingly important for the companies and the businesses in which they operate. By using the multiple-case study approach, we were able to identify common activities that different roles in embedded systems companies perform in parallel and concurrently when developing, deploying and evolving ML/DL models. Based on this understanding, we derived a framework to incorporate ML/DL into the larger business context of a company.

5.2.1 Case Companies

In this section, we present the six case companies involved in our previous as well as current study. For each company, we describe the different ML/DL use cases that we studied. All the reported use cases are using real-time datasets to build ML/DL models in real-world settings. Table 1 provides a detailed description of each case company as well as the experts involved in the study and their roles within the company. In this table, case company A, B and C are part of the previous study while companies D*, E* and F* are part of the recent study. *I* denotes Interview participants

and *W* denotes Workshop participants of the the study.

Case Company A - Telecommunications: A telecommunications company with a multinational network of partners that includes e.g. vendors, operators and customers. In case company A, we studied four use cases in different business areas.
A1. Log Analysis: With the advent of ML/DL technologies, log data can be used to gain valuable insights. Logs generated by different products are retrained to generate profits for the company by reusing all existing infrastructure, models and parameters. The dataset consists of gigabytes of log files generated by different products within the company.

A2. Paging Cell Phone: To access a specific cell phone, there is a need to page one or more base stations. The company analyzes the mobility patterns of user equipment (UE) within the network to more specifically page the mobile phone while it is in idle mode using an existing paging algorithm. The dataset contains all events that occurred for a specific node in the company (approximately a thousand gigabytes of data) within one week.

A3. Detecting Garbled Speech Frame: Speech frames are encoded when voice data are encoded over audio. Certain flags in a specific speech frame contain information about the previous and subsequent frames. The company predicts whether the speech frame is garbled or not by analyzing this information. The ML model should be super-fast and accurate to be placed in the system. The dataset consists of speech frames collected from the company.

A4. Predicting Hardware Faults: The company predicts hardware faults intending to minimize the amount of hardware returned by customers for repair. The information collected from crash log reports of different customers helps in the building of the ML model. During hardware screening, they focus on two aspects a) if the hardware is found to be faultless, it is returned to the customer b) if the hardware has faults, it is sent to the repair centre. The dataset consists of crash log reports collected from the company.

Case Company B - Automotive I - *Autonomous Driving Vehicles*: A

company that manufactures trucks, buses and construction equipment as well as supply marine systems. The vehicle needs faster DL algorithms when driving on high-speed roads. Furthermore, the company needs to ensure that the rate of failure is minimal for these safety-critical products. The company develops DL models in collaboration with external partners. The dataset consists of millions of manually generated test cases based on accident statistics or based on the fabrication of typical traffic situations, which are then mutated or combined in various ways. Furthermore, it contains thousands of hours of data recorded while driving in real life.

Case Company C - Packaging - *Defect Detection*: A company that provides processing and packaging solutions for food and beverages. The company utilizes DL models to detect defects in packages such as dents, wrinkles, etc. at each customer site during processing. The global model in the cloud is trained using learnings from local training at each customer site. To detect customer-specific defects more effectively, the learnings at the cloud are then pushed back to the customer sites for inference utilizing transfer learning. The dataset consists of packages with different patterns, types and colours.

Case Company D - Automotive II - *Object Detection*: The company acts as an innovation centre that provides mobility solutions for passenger cars. The company focus on redefining automotive engineering by introducing smarter solutions. The company utilizes DL models to detect forgotten objects by taxi passengers. These DL models are used by the company to detect items such as keys, wallets, etc in the taxi. The use case has proven to be worthwhile because collecting and storing these forgotten objects is quite expensive. The dataset consists of a large number of images with forgotten objects inside the taxi.

Case Company E - Pump Supplier - *Water Supply Rate to Pumps* : The company manufactures different kinds of pumps as well as electronics for its control. The company has extensive networks in many counties and within these countries, they distribute products with the help of local distributors. The company designs and optimizes pump solutions to meet the needs of its customers. When there is an excess of water, the company uses ML models to provide insight into how much water flows into the pumps. The dataset contains both mechanical and operational data from

pumps.

Case Company F - AI Development Platform provider - *Audio Analysis* : A company that provides a platform for AI model development and deployment. Teams in their customer companies can easily operationalize AI with the help of this platform and increase business. The main goal of the company is to make AI accessible to everyone so that they can concentrate on the problem and avoid spending time on repetitive coding tasks. The company develops DL models to distinguish audio analysis for industrial predictive maintenance. The DL models are used in industrial sites to identify early defects with machines. The dataset includes both good and bad audio from healthy as well as unhealthy machines.

Table 5: Description of case companies and practitioners involved in interviews and workshops

Case company	Description	Experts	
		ID	***Roles***
A	A company providing software, services and infrastructure in communication Technology	I1	Senior Data Scientist
		I2	Senior Data Scientist
		I3	Instigator
		I4	Data Scientist
		W1	Data Scientist
B	A company manufacturing and marketing vehicles	I5	Technology Specialist AI/ML
		W2	Director
		W3	Manager
		W4	Software System Architect
		W5	Research Engineer
C	A company offering packaging and processing solutions for food products	I6	Data Science Manager
		I7	Solution Architect
		I8	Data Scientist Analyst
		I9	AI Application Specialist
		W6	Head of Data Science
		W7	Director
		W8	Manager
		W9	Technology Specialist
D*	A company providing mobility solution for vehicles	I10	Vice president
		I11	ML/AI Engineer
		I12	Data Scientist
E*	A company manufacturing pumps and electronics for pump control	I13	Project Head
		I14	Senior Data Scientist
F*	A company providing platform for AI development	I15	Senior Data Scientist
		I16	Head of Research
		I17	Chief AI Officer
		I18	Senior Data Scientist

5.2.2 Data Collection

The research reported in this paper builds on a semi-structured interview study conducted between September 2019 - May 2020 to collect qualitative data. With an increasing understanding and knowledge of the development,

adoption and evolution of AI/ML/DL technologies, the first author developed the interview protocol. Based on feedback and recommendations from other two authors with great experience in AI/ML/DL technologies, the first author added some additional questions, merged similar questions and removed some irrelevant questions. The interview protocol consisted of two parts where Part I focused on the role and experience of the interviewee and, Part II focused on current development practices and activities related to the development of ML/DL models and explored the challenges experienced by data scientists in the design of ML/DL models. We provide the interview protocol covering Part I and Part II in an Appendix section (Appendix A). Based on the study objective, we planned the study by finding key contact person from six case companies that are part of a larger research initiative based on their experience in dealing with AI/ML/DL projects. We did snowball to identify practitioners with experiences in developing, deploying and evolving ML/DL models from each company after receiving suggestions from these key contact person on practitioners suitable for the study. Once we identified appropriate practitioners, we sent a personalized invitation to the interviewees (key contact person + interested practitioners) and agreed on a suitable time slot. As a result, our interview study involved 18 experts from six large companies representing different business areas and domains. The practitioners involved in our study has wide range of experience between two to over ten years. All interviews lasted one hour and were conducted via video conferencing. With the consent of the interviewees, all interviews were recorded and transcribed for analysis. At the end of each interview, an opportunity for follow-up questions was agreed upon. In addition to the interviews, and as part of the overall research initiative, we continuously met with the case company representatives in different workshops, meetings and events organized by both the case companies and the researchers at the universities to collect secondary qualitative data. We organized a total of five workshops in which we visited the same case companies to present our preliminary findings based on the study, obtain their feedback and better understand their ways of working. These workshops served as an opportunity for practitioners to discuss, share insights and experiences as well as for us to get different perspectives from a group of people interested in ML/DL models. Furthermore, the workshops provided excellent opportunities to explore many of the challenges associated with the development of ML/DL models. In addition to interviews and workshops,

we also conducted several meetings and events with representatives from the case companies to improve the quality of our research study. During those meetings, we obtained additional knowledge on their current work practices, their overall system development process and their business and its visions and goals. Furthermore, the AI-Driven Business Development Framework illustrated in Figure 2 was presented at these meetings and events for validation.

5.2.3 Data Analysis

During analysis, each interview transcription was carefully read by the first author and summarized and discussed with the other two authors. Also, notes were taken during the interviews to help summarize the key content. Also, any potential question or potential misunderstanding was discussed with the respondent in order to avoid misinterpretations. In addition, we took notes when presenting our preliminary findings to the practitioners at follow-up workshops and when validating the AI-Driven Business Development Framework in Figure 2 at the meetings and events. Our notes, as well as the feedback we were given by the company practitioners, were continuously included in the analysis process. We applied elements of open coding to analyse the interviews, workshops, meetings and events in order to identify key concepts and subsequently group these into conceptual categories that captured the development activities performed by the companies, the roles and functions involved and the many challenges they experienced along the way of developing, deploying and evolving ML/DL models [97]. Triangulation enhances the precision and validity of our study [98]. Since we were three authors conducting the interviews and reading the interview transcripts, notes and summaries, triangulation provided multiple perspectives on the subject under study [99]. The first iteration aimed to identify additional empirical knowledge for generalizing our previous study findings. The second iteration was performed to identify high-level activities that companies undergo when developing, deploying and evolving ML/DL models. The third iteration was carried out to identify the challenges in designing models. In the fourth iteration, we identify iterations and triggers between phases to optimize ML/DL development as well as the roles and organizational functions involved based on the study. The final iteration was carried out to assess different decision points for immediate termination of less value business cases. The results

deduced from the analysis were sent to the practitioners involved in the study for validation.

5.3 Empirical Findings

We report on the empirical findings from each of the six case companies involved in our study. Companies adopting ML/DL technologies are interested to know if they are useful/suitable for their business context or not, cost-effective and reliable for them, how much it will cost to set up necessary infrastructures, valuable data collection procedures as well as the development, deployment and evolution of models. These are currently not fully investigated in the company or academia, leading to difficulties in determining the suitable business case for the adoption of such technologies in the company [127]. As a structure for presenting our empirical findings, we organize it into three viewpoints: business, data and models. Adoption of ML/DL technology in business is futile if it generates no benefit to customers. High-quality data is required to train models that can detect hidden valuable patterns in the dataset. The value of a particular ML/DL use case can only be realized by operationalizing the ML/DL models. We believe that this categorization helps to give an overall idea of the different elements that companies need to work on, and advance in when moving towards an AI-driven business. Below, we describe the business opportunities for AI at each of the companies and their organizational structure as well as how they deal with data and ML/DL models.

5.3.1 Business, Data and Model opportunities in case companies

Case company A: Telecommunications:
Business: Case company A offers products, software, services and infrastructure in information and communications technology to service providers. The company adopts AI across services and products to enhance automation, advance already existing products, build new business opportunities, increase revenue streams and improve customer experience. Inside case company A, the typical use cases for ML/DL are the ticket routing of trouble reports, classification of faults in the logs, etc. Company A consists of experimental prototyping or half applied research teams in addition to product development teams. They focus on proof of concept or pilot style projects for the product development teams. In company A,

R&D is organized in a DevOps fashion with developers and operations (e.g. release) in close collaboration and cross-functional feature teams.

Case company A implements ML/DL to a product if it adds value to customers compared to the implementation of traditional approaches, reduces human effort and saves time. When profitable business cases are proposed, we observe that data scientists spend one or two days running simple ML/DL models over a structured dataset to verify whether or not business case criteria can be met. They schedule discussions with product owners to see if the accuracy achieved is worth it or to invest time enhancing accuracy. On the other hand, certain business cases can only be resolved with ML/DL to generate value, even though they entail high costs. For instance, building an infrastructure for collecting and examining radio data for the implementation of an analysis tool entails high costs.

> I3: *"It is very hard to nail down exactly the business case when it comes to ML/DL because typically there are a lot of costs involved than ordinary projects."*

Business cases are validated by customers based on the end goal requirements before proceeding to deployment. Requirements typically include accuracy, prediction time, available computational resources, understandability, interoperability with other environments, reproducibility, model execution environment, etc. For instance, in the case of detection of garbled speech frames, accuracy has been given greater importance because the misclassification of the correct frame is much worse than not classifying the incorrect frame. In a different scenario, ML analyzes customer trouble reports to identify the team to which the report should be forwarded. It will take a while before an individual examines the report when it is generated. Prediction time is not important in this case.

Data: In case company A, data scientists with the help of developers, generate data that can be used for ML/DL business cases. Data scientists find it very difficult to collect data once they sell a product to a customer who owns the data. In such cases, they must either ask customers directly or try other methods to obtain valuable and representative data. For instance, as access to node configuration is restricted and data is anonymized, data scientists spend three months analyzing dataset events to reverse engineer the baseline. Data scientists spend time with domain experts to understand the data and the system. They use visualization techniques for data ex-

ploration. They either use domain knowledge experts to label the data or use unsupervised or clustering techniques for labeling. One of the data scientists suggests that it would be a good idea for domain experts or data scientists with the domain expertise to sit down physically with the team to gain an understanding of the data. This seems to be a major problem in the company as it is complex and involves a lot of interdependencies. As a result, data scientists manually chase and contact domain experts once they have access to the data. Data scientists continuously build and test hypotheses around the data to better understand the data and the system. It also offers insight into non-functional requirements, such as computational budgets, model expectations, etc.

> I2: *"We always start with data, look at the data and just explore the data as much as physically possible. Just to understand it and to get some intuition around what is going on and what is behind all this data... It is highly unstructured in the way that we do not know where the data will take us, so we try building and experimenting with the hypothesis that we have around data."*

The majority of data scientists compose features with the help of domain experts by using two approaches: a) Start with a high-dimensional feature set and scale down to a few-dimensional set by removing irrelevant features that do not affect model performance b) Scale from a low-dimensional to a high-dimensional feature set by adding relevant features. Data scientists often add features that are not explicitly part of the dataset to the feature set based on the domain knowledge perspective.

Model: Data scientists follow certain guidelines for the implementation of the ML/DL models. They use classic ML techniques for smaller datasets, DL for large labelled datasets, and clustering techniques when clusters are visible during data exploration. Some data scientists apply random forest to small dimensional problems and pairwise correlation between features for relatively small dimensional problems. Typically, data scientists take a small sample from a large dataset and identify the class of problem as if it is an image/text etc. They choose two to three approaches, ranging from simple and classical to more advanced approaches, such as DL, that have traditionally/historically performed well in the identified problem class.

> I1: *"There is no scientific precise method for decision making. It is completely based on from one problem to another problem. That is not well-defined. I don't think anyone can say, "yes, this is like a formula that you can use"...So the key is just to have a scientific and objective approach, I think that is probably the ticket rather than just having the formula."*

We note that data scientists are interested in automating experimentation tasks using tools such as H2O.ai, AutoML, Auto-WEKA, etc. They have different attitudes when working with ML/DL models. Some data scientists do not care how the model works as long as it works, while others use accessible domain knowledge on simple models, which contributes to an easily explainable model. To optimize the model, data scientists adopt hyperparameter tuning that can be based on previous literature or on any findings or experiments that data scientists would like to make. It is difficult, according to *I3*, to significantly increase performance by experimenting with another model after the finalization of a particular model. Experimenting with the model that has already been finalized is a good choice in this situation.

By introducing ML/DL into the company, data scientists and ML engineers focus more on developing and deploying ML/DL models. Data scientists put requirements on ML/DL models before proceeding to deployment to ensure a code review, unit testing of all components, ready-to-use deployment infrastructure, proper review of model training, check whether or not a model build by a data scientist is understandable to another data scientist or not, maintenance, robustness, and stability testing. Experts focus on three aspects to bring the model into deployment and integration. First, they prepare code ready for easy deployment in the docker container. Second, a plan for integration with existing internal systems in the company, and, thirdly, use reusable functionality existing in API services to wrap up the model to provide services to business owners. According to *I1*, despite huge investment in AI and data scientists in the company, the transition from prototype to production quality models is quite slow. Data scientists place the model under close supervision of A/B testing. They also use a model execution environment with built-in support for A/B testing, canary selection, etc. as an in-depth part of the product. Deployed services include a training interface, inference and evaluation interface to train data, retrain with new data and compare new models and old models

to choose whether or not to roll back. Data scientists often adopt a strategy where the old model has been adapted with new functionality, resulting in a newer version of the model placed in the same infrastructure. Then compare between the old and new model to choose the best model. In contrast to this approach, data scientists perform continuous retraining by adding functionality to the existing model. Data scientists carry out continuous monitoring and logging activities to ensure that the model performs as expected.

> I3: *"Maybe what is a bit specific putting ML/DL models in production is that you want some way to monitor that the model keeps performing the way you need it to."*

Case company B: Automotive I:
Business: Case company B is engaged in the design, production, and supply of automobile vehicles as well as the use of ML/DL technologies for predictive maintenance, image and speech recognition, etc. The company considers automation to be the key aspect of AI adoption, which in turn generates customer satisfaction. AI is primarily used for perception in case company B. Since autonomous driving entails a lot of privacy and security concerns, vehicles will be hit by the road after a lot of V&V (verification & validation) activities. The company relies on external collaboration for the development of product-level components in the case of autonomous driving vehicles for public roads.

In case company B, the software development organization practices an agile way of working. Practitioners work together in a team, across the teams and in the company network. During the initial outbreak of AI techniques, the company started an ML team. We note that a single person is responsible for various roles in the company. For instance, *I5* acts as product owner for two different teams and as project manager for other projects. The company relies on running advanced engineering or research projects for autonomous driving. Data scientists in the company use DL for perception and a mixture of ML/DL for other projects. Case company B confirms the need for established processes compared to the existing development model and processes at a later point in time.

> I5: *"I think that question is very much dependent on who gets it and what their context is...Since the company is big, we have many different departments, contexts and application areas.*

But for us who work with autonomous driving, it is a very immature area which is research-oriented where everyone is frantically trying to build products."

Data: Compared to other case companies, case company B needs to collect huge data for V&V (verification & validation) as these activities are necessary for safety-critical products. As a result, the entire design of the autonomous driving vehicle, along with KPIs (Key Performance Indicator) and methodology, is based on V&V. For instance, V&V is important because there is a high probability of killing or injuring people while driving vehicles on the roads. According to *I5*, data collection is not hard if sufficient resources are provided. On the other hand, experts have varying opinions when it comes to perception problems. DL models require a large dataset compared to ML models for better performance. For instance, datasets of safety-critical products are collected by manually building, mutating and simulating test cases, and recording thousands of hours of data while driving on the roads to prove that failure rates are low. Verification and validation activities are significant for the design of ML/DL models in the case company. For instance, there should be a greater degree of certainty that the system performs well in the case of safety-critical products, as it is likely to injure or kill people. The company depends on a third party for data annotation. For instance, the company builds and drives vehicles, while a third party provides the infrastructure to access the data.

I5: *"Need enormous amount of data for perception systems, so it is hard to collect all that data...We need to prove that the failure rates are low enough."*

Model: According to case company B, there are specific ways of working with ML/DL and non-ML/DL systems. Data scientists focus on a state-of-the-art approach to find the current best algorithms for research-oriented projects and well-known and explored algorithms for a deeper understanding of small-scale pilot products. For instance, as object classification is a key issue for autonomous-driving vehicles, data scientists look for state-of-the-art and select the best available algorithm. They also conduct a feasibility study before selecting a particular algorithm. As per case company B, high AI and DL knowledge are required in case of perception. The knowledge involves understanding how to combine networks, how to work confidently with the performance they provide, limitations and

the integration of individual DL models into a well-functioning ensemble model. For established products such as vehicle perception, the data scientists make realistic architectural decisions in terms of computational speed, memory speed, real-time execution, etc. For instance, select an algorithm that matches the performance of the hardware available during the lifespan of specific product development. In safety-critical products, experts apply different method levels to the ML/DL component to ensure proper system behaviour. For instance, they apply ISO 26262 and SOTIF (Safety Of The Intended Functionality standard). Data scientists find the ML/DL project challenging if the product itself changes frequently. For instance, changing algorithms would result in higher costs as huge validation activities for safety certification are required in autonomous driving. Upon completion of the required V&V, the DL models are mounted onto the vehicles for perception.

> I5: *"It costs a lot of money to change the algorithm in such a large project because a lot of validation is going on in the case of safety-critical products."*

Case company C: Packaging:
Business: The case company C produces high-quality packaging, processing and filling machines for various food products. AI has been adopted in the company to enhance automation to increase productivity and decrease costs. The company provides customer satisfaction, brand exposure as well as protect the taste of the food product. Typical use cases in case company C are detecting defects in fully finished or semi-finished packages, a quality check of sealings, detecting food misplacement, etc. Detection of defects by mounting a camera to the production line improves value delivery. Triggering warning during defect detection saves significant time and costs. The company also makes an effort to apply for patents on its valuable and novel inventions in order to improve its market position and prevent price competition among competitors. In comparison to the majority of case companies, case company C has a more mature level of ML/DL adoption. They develop appropriate architectures and frameworks for the implementation of ML/DL business cases with the assistance of solution architects in the company. The company relies on the digitization of food manufacturing in close collaboration with external partners. A data science team in the case company typically consists of a data science manager, senior data scientists, junior data scientists and solution archi-

tects. According to data scientists at case company C, the company will benefit from having both productions and research focused teams operating concurrently. Productions focused teams rely on the traditional cloud approach while research-focused teams experiment with concepts related to edge (re)training.

> I6: *"Some teams in company perform standard tasks while others try trending concepts."*

Data: Case company C already has mechanisms in place for cloud (re)training and edge inference. They intend to move from this centralized approach toward a a fully decentralized approach. The company has a global model for (re)training in the cloud and a local model for (re)training at the edge. The dataset needed for global model is collected either by direct sampling or through the manual dataset generation while the local data set is unique to each client-site. The company occasionally employs interns for dataset labeling. To increase global model performance at cloud, mislabelled data after edge inference is updated to the global data set. As the company moves toward decentralised approaches, appropriate decisions have to be made concerning the amount of data to be transferred to the cloud, how to deal with labeling, feature selection, quality issues, model compression, Type1 and Type 2 errors, etc.

Model: We note that the company take the initiatives to schedule frequent meetings and discussions between ML/DL practitioners and non-ML/DL practitioners to bridge the communication gap between them. For instance, architects in the company are provided with end-to-end knowledge of the entire ML/DL business case. The company looks for published papers to find state-of-the-art learning. Before finalizing a specific algorithm, *I8* experimented with available algorithms suitable for the business case. They also participate in conferences and workshops to further advance their knowledge on the recent advances in the field. The global model (re)trained at cloud is deployed to edge if it outperforms the local model. The model performance is evaluated using metrics such as mean average precision, accuracy, false positives, false negatives, etc. To reduce power and resource requirements at the edge, the global model is compressed before deploying at the edge and later decompress without diminishing accuracy. To optimize model performance, the company tries to employ techniques such as transfer learning, federated learning, etc. to deploy

models at the edge. The company ensures model management by sending logs to the cloud when errors (Type 1 or Type 2) occurs.

> I8: *"The simplest part is deploying AI on the edge, while most challenging part is evaluating global model retraining."*

Case company D: Automotive II:
Business: An innovation company dedicated to providing smarter mobility solutions for automotive. Based on AI, the company attempts to strengthen its capability to innovate and add value to products. Most business cases emerge as a result of customer requirements or as part of in-house research initiatives. In the case of research ventures, the UX (User Experience) designer or the innovation management in the company determine if the business case is valuable to any of their business owners/customers. If it is valuable, the company transform the business case into a real-world project for them. Once the business case is confirmed, the company sets up data science teams to improve future mobility. A typical data science team in the company consists of product owner, data scientists, solution architects, etc. If a business case delays or fails due to insufficient data, data scientists explore what more can be done for the business case in the future. Unfortunately, they will not be able to apply their expertise in the current situation and only be involved in the preparation of future projects. The company extends the number of data scientists to apply ML/DL technologies to more valuable use cases. If the proposed business case does not provide value to business owners from the outset, it should be terminated as soon as possible. It is quoted below:

> I10: *"From the first idea if there is no business case, you stop"*

Data: To label data, the company employs MTurk services. Since people from all over the world contribute to the annotation process in MTurk, confidence is always low. The company integrates data version control (DVC) into the workflow. When annotations are initially bad, they can refine and gradually update and then use that data to train models. In case company D, DVC and GIT (Global Information Tracker) work cooperatively. During scheduled meetings, data scientists encourage business owners with less ML/DL knowledge to complete a series of questionnaires. These questionnaires can bridge the communication gap between data scientists and business owners. For instance, the questionnaires include questions such

as what do you want to answer? what kind of data do you need to respond to this? do you know of any domain experts who might be able to tell you what kind of data you need to address this? and so on. Most business owners believe that data scientists understand their business and can formulate questions and answer their concerns, but this is not the case with the vast majority of data scientists. Often, data scientists aim to clarify misconceptions about AI and convey to business owners that if valuable relationships can be extracted from their data, they can automate the decision-making process for them. According to *I10* and *I11*, data exploratory analysis is dependent on projects.

> I11: *"The specifics of how you do it, what you are going to look at it, depends a bit on the project. But I think the overall reasoning that you have to do exploratory data analysis is very much there."*

Model: The company is still in its early stages of ML/DL model development, with very little integration with already existing systems. They also rely on state-of-the-art to identify trending algorithms in their domains. We find that there is a slight change in the conventional wisdom of the small versus large model in the company. For instance, when dealing with text data, it seems that more blog posts, articles, etc. suggest probably using a very large model and train it for a small amount of time and that turns out to have better results. Some of the projects are delayed for deployment in the company. According to data scientists in the company, even if a lot of algorithms are available in publications, very few have working code that can be experimented with. So there is a high probability that publications without working code gets rejected if the business has to be implemented in a short time.

> I11:*"Since we do have some time pressure on how to do what we want, we have taken the kind of publications that had working code and we have looked at it."*

Case company E: Pump Supplier
Business: Case company E is a pump manufacturer as well as a pump retailer. To promote real-time monitoring and fault prediction of pumps, the company employs AI, digital services and cloud networking. A particular use case for the company is the condition monitoring of pumps

and other critical assets to identify the need for maintenance and repairs. Traditionally, product teams develop, maintain and operate services in case company E. The company is attempting to digitalize the provided service to enhance efficiency and this idea of digitizing a specific service is initiated by a business developer. For instance, a business developer proposes the concept of a wastewater network in the company. This will enable wastewater utilities to monitor the condition of pumps when there is an influx of excess water. The case company E discusses various business cases at their internal ML network meetings. A typical data science team in the company includes data scientists, data pipeline owner, domain experts and business owner. Other typical roles include back-end or front-end software developers, subject matter expert groups, etc. One of the data science teams in the company consists of 10 experts and students, while the other team consists of 12 data scientists and 8-10 data engineers. The company has cross-functional teams, which allows them to reach out to different teams for the development of ML/DL models. For instance, the data pipeline team collaborates with the marketing team because they both have data scientists and data engineers who understand data and its purpose.

> I13: *"And then we support the teams with some subject matter expert groups, some of them are for software development like DevOps, some of them are from my team - data pipeline and platform. They support on how to set up data architecture or tools to use to set up data pipelines and how to make good data quality measures"*

Data: The company sets up devices to collect data for different business cases. They often tend to begin with a small dataset because beginning a project with a large dataset is very costly. Following the collection of data, the company uses a quality assessment method to conduct a structured dialogue about data quality dimensions of the dataset. According to *I13*, data scientists in other companies ignore data dimensions when using software development methods. For data exploration, data scientists collaborate with domain experts. They often start with small datasets for investigation in the alpha phase of the project. According to *I13*, the key difference between DevOps and DataOps is that it puts a greater emphasis on data architecture rather than technical architecture. Data scientists confirm that training-serving skew causes models underperform in production.

For instance, the typical format, alignment and timing of the dataset vary from the training set to the training set.

> I13: *"So we have to rethink all the way how you can develop your in-house solutions because that is what I see when we use the software development methods, we disregard the data dimension and then we see all kind of problem coming up all over the place."*

Model: To introduce service digitization in case company E, they adopt/explore state-of-the-art AI, ML and cloud computing technologies. The company relies on a literature study to identify suitable statistical methods for the business case as well as relying on business owners to provide a fair idea of KPIs. The case company has tools for testing and deploying models, easy switching between models and handling data lineage. According to case company E, the old model must be decommissioned when a new model is created and deployed in production. Employing agile development, DevOps, data science, and MLOps makes the development, deployment and maintenance of models more easier in case company E.

Case company F: AI Development Platform provider

Business: By providing a platform for building and deploying systems, the company aims to simplify AI development so that its benefits are accessible to non-experts. They assist customer companies in their transition from analog to digital by assisting them in AI adoption. The case company F will investigate relevant potential issues before finalizing the business case. It offers an AI platform lifecycle management for increasing productivity and helps business owners in solving their problems by offering a DL development platform. Aside from the main data science team that works with the business owner, they have research teams that look at emerging techniques, for instance, federated learning. When the business case is finalized, data science teams are formed within the company as well as junior and experienced data scientists are paired up when working on projects. The data scientist working in a specific business unit should become familiar with the field and eventually become a domain expert. In contrast, they are not interested in spending time worrying about it, but rather want to become quickly acquainted with the field. Since the number of data scientists is small, the company believe it is important to teach data science to software engineers to increase value delivery.

> I17: *"Defining exactly what should be the problem before going into the project and work for few months is much more important than you think."*

Data: The company plans early meetings with business owners to obtain a more comprehensive understanding of the business case. These meetings aim to bridge the gap between lack of knowledge and understanding of how ML/DL works among business owners and how the domain works with data scientists. These activities can also help to avoid a scenario in which the problem appears to have adequate data and prediction appears to be reasonable, but the difficulty is that there is no business case to justify the model development. Building a fraud detection system by collecting a vast amount of data and spending hours, for instance, tends to be futile if fraud happens only once a year. For the data collection process, data scientists request assistance from domain experts. They conduct a quality test on a sample dataset in the early stages of the data exploration to assess which data can and cannot be used for the business case. According to data scientists in the company, the key difference between ML and DL is that conventional ML needs a substantial amount of work in feature engineering. Since data scientists are costly, their efforts can utilize in other valuable tasks when working with DL. When relevant features in ML are removed, there is a high risk of information loss. By using DL instead of ML, we can avoid errors that occur when people are biased during feature engineering. On the other hand, feature engineering in ML is useful in two ways: a) Reducing dataset and finding good data points from a noise point of view to remove unclean data, b) Simple algorithm does not handle a large number of features.

> I17: *"DL Models are so complex that they can learn if you have enough data"*

Model: Data scientists rely on a literature study to find state-of-the-art for developing DL models. To achieve model optimization, data scientists a) adopt hyperparameter tuning, b) Introduce more variances to the dataset, c) Fine-tune the annotation, d) Regularize weights e) Understand part of the data for which the model underperforms. The robustness of the algorithm can be enhanced by artificially introducing more variances to the dataset. For instance, add variation to the input images by cropping or moving them around to create previously unseen images. Reconsider the annotation as

there is a good risk that it is causing the model to underperform. In such cases, refine the annotation and eventually update the data used to train the models. In the case of weight regularization, the network's weights are reduced to a small value if they do not contribute to the model performance. If a model developed based on prior work fails, data scientists may verify the discrepancies between the two datasets as an initial step. They try, if possible, to validate the business case using the dataset described in prior works. If the problem still persists, they experiment with new algorithms. Mainly, data scientists believe that business owners have sufficient resources to do the integration and equipped with interfaces to input data and obtain results. In contrast, in most cases, the data science team must set up the first time integration for business owners. This may be due to a lack of resources within the company or a lack of a diverse engineering organization or the possibility that they are busy with other tasks.

> I16: *"Even though we have a lot of experience having those meetings and even though we know many of the problems, we tend to fail. Even if we set up a problem then also we miss things that we have not communicated well on expectation"*

> I17: *"But I think we have evolved a lot and I think we are getting to a point that we need to solidify on a set of best practices but we have not yet."*

5.3.2 Challenges

During our study, we identify several challenges that company practitioners face in their day-to-day practice of developing, deploying and evolving ML/DL models. Below, we provide a summary of the key challenges that data scientists experience and are common among companies. The challenges are grouped into three categories related to ML/DL model development: a) Pre-Deployment, b) Deployment and c) Non-technical challenges. We detail each challenge and we provide an illustration in Figure 1 and we provide the frequency of these challenges identified in each company in Table 2.

Pre-Deployment

1. **Representative and Valuable Dataset**: The majority of data scientists involved in our study confirm that DL models need a large dataset than

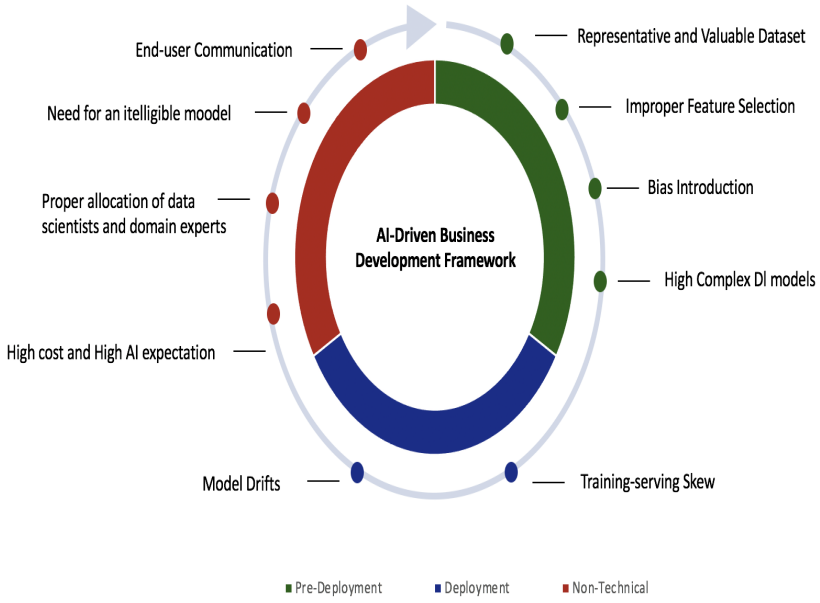

Figure 5: Challenges in ML/DL Model Development

ML models. For instance, to show that failure rates are low, datasets of safety-critical products are collected by manually building, mutating and simulating test cases and recording thousands of hours of data while driving on roads. In contrast, one of the case companies disagrees and reports that there are several misconceptions about DL models, such that as the need for a large amount of data. Mostly, the collected data will be noisy, have few labels or be completely unlabeled. According to data scientists, the dataset collected for ML/DL must be a valuable and representative sample. For instance, to classify prescriptions containing antibiotics, a dataset with data points for prescription containing and not containing antibiotics is required. Furthermore, in order to determine whether an industrial process is operating normally or not, the dataset requires data from both sites where the process is operating normally and from sites where it is not.

> I10: *"If I want to do machine learning, I will also need negative samples. Most datasets are skewed toward abnormality, this is usually not a deal-breaker, it is important to keep in mind early on."*

2. **Improper Feature Selection**: Most data scientists consider feature selection to be an important step and indicates that a feature set with insignificant features has an impact on model performance. Without a comprehensive understanding of each feature, data scientists struggle to add relevant features to the feature set. For instance, if two features contain the same information, it is difficult to choose the significant feature.

> I4: *"Adding a new feature is costly in the way we work. So, we want to have an idea of what this feature can bring us before we do the implementation."*

3. **Bias Introduction**: We observe that data scientists introduce bias based on their experience in algorithm selection for both ML/DL models as well as in the case of feature selection for ML. For instance, most data scientists prefer to use näive bayesian models, logistic regression, tree models, random forest and support vector machines as favourite models to achieve base performance. We notice a conflict in the study when one of the data scientists refer to the random forest as being computationally fast to train and produce results in seconds, whereas another data scientist indicates that random forest is flexible but less interpretable.

> I1: *"They bring their own experiences and then introduce own biases into the whole working."*

4. **High Complex DL Models**: Despite the popularity of DL models, most data scientists prefer ML models for training because they are less complex. In comparison to ML, they believe that deep knowledge is required for the implementation of DL models. For instance, in the case of safety-critical use cases such as perception, deep DL knowledge is necessary to understand neural network limitations, network merging, activation function, etc.

> *"We need to have a lot of AI and DL knowledge to understand the limitations of the networks, when they may fail a lot ..... We are only half done after we have trained the networks."*

Deployment

1. **Training-Serving Skew**: Data scientists consider training-serving skew as the disparity between training and serving performance. For instance,

the system is trained on events to ensure that it performs well in safety-critical products, but it is less suited to real-world situations while serving.

> *"It is very hard to capture real data for these long-tailored events because some of them will happen so rarely."*

2. **Model Drifts**: Data scientists perceive model drifts as a potential threat to model performance. As a result, it is necessary to properly detect model drifts and determine when to retrain or update. Models, for instance, are retrained on a daily, weekly, monthly or year-to-year basis, depending on the domain and when input data changes.

Non-technical

1. **High Cost and High AI Expectation** According to the study, the implementation of ML/DL models requires significant infrastructure costs when compared to traditional software development. According to case company A, experts with little or no experience in data science have high AI expectations, so it is feasible to gain early insight into what can be achieved in a specific business case. For instance, when product owners hear the term "AI", they have extremely high expectations. Unfortunately, data scientists are mostly subjected to high-level pressure to solve business cases using ML/DL. The majority of data scientists agree that it is difficult to formulate a business case for business owners who have a lot of data but no knowledge of AI.

> I4: *"We are assigned the business case, but we never get any information about the quantitative specifications to be optimized.....To make good business value, we need input from stakeholders, and sometimes we need to play an educational role because they are not aware of the potential risks involved in the design of ML/DL systems."*

2. **Proper Allocation of Data scientists and Domain experts** Most experts involved in our study confirm that data scientists and domain experts must be carefully allocated to the project. Since data scientists are scarce resources, they need to be assigned to projects of high business value. For instance, we discover that the number of data scientists in case companies is significantly lower than the number of other technical staff. Data scientists, on the other hand, find it harder to consult domain experts in a larger

company due to their small number. We note that this may be because companies have domain experts allocated to several ML/DL projects at the same time. We also notice that one of the case companies relies on interns, particularly students, to add labels to data. Even though one of the companies uses a labeling tool to provide data labeling, reviewing these labeling is a tedious task. It is also difficult to find optimum models due to a lack of adequate data scientists to perform hyperparameter tuning.

> I1: *"Even then as a data scientist, you need a domain expert to label it."*

3. **Need for an Intelligible Model**: Most data scientists highlight the importance of understanding the model developed by other data scientists. In the worst-case scenario, they spend a significant amount of time understanding the model, which slows down the overall project speed.

> *"It is very hard for another data scientist to understand the model in the case of some models that we have taken over from a data scientist."*

4. **End-user Communication**: All data scientists report that there is a need to encourage more communication with end-users because the model is difficult for them to understand. For instance, data scientists answered all of the questions raised by customers because they lack strong confidence in AI in the case of the analysis tool designed to analyse radio data. In other cases, data scientists collaborate with testers who have no prior experience in developing ML/DL models. Since these testers struggle to understand the models, data scientists need to spend time explaining the models to them.

> I4: *"We need to educate them because they are very interested in understanding why we have done some specific predictions."*

> I3: *"If you take an ordinary software component no one questions it as it is conceptually quite understandable; but with ML/DL models, there are many more questions."*

In Table 2, we classify the occurrence frequency of challenges common to six case companies into three divisions. They are High (H), Mid (M)

Table 6: Occurrence frequency of challenges in case companies

Categories	Challenges	Case Company					
		A	***B***	***C***	***D***	***E***	***F***
Pre-Deployment	Representative and Valuable Dataset	H	M	L	H	H	L
	Improper Feature Selection	M	L	L	M	M	L
	Bias Introduction	M	L	L	L	L	L
	High Complex DL Models	L	H	M	H	L	L
Deployment	Training-Serving Skew	H	H	H	L	L	H
	Model Drifts	H	H	H	L	L	H
Non-Technical	High Cost and AI Expectation	H	M	M	H	H	L
	Proper Allocation of Data scientists and Domain experts	M	L	L	H	H	L
	Need for an Intelligible Model	H	M	M	H	M	H
	End-user Communication	H	H	H	M	H	L

and Low (L). As can be seen in the table, the majority of companies face deployment challenges. This may clarify why there is less transition of ML/DL models from prototype to production quality in case companies. Companies with very few deployment activities believe that the occurrence of these challenges is very low. Most companies think that a dedicated team should be set up to monitor deployment issues. End-user communication is a challenge for all companies when they need to interact with customers, testers, developers, architects, and others who have little or no ML/DL knowledge. It should be noted that case companies working specifically on DL algorithms, see fewer instances of improper feature selection and bias introduction challenges since DL models learn patterns directly from data. In some companies, the availability of experienced data scientists is still a nightmare.

5.4 Discussion

In this section, we discuss our empirical findings and we present a framework that was inductively derived from our empirical findings. Based on the empirical findings, the framework gives a better understanding of how the adoption of ML/DL technologies generate profit to companies and how it can be integrated to deliver high business value, different activities performed by practitioners, their ways of working with business cases, data and model as well as iterations and triggers to optimize model design and roles and organizational functions involved. The framework provides an end-to-end conceptual framework to ensure continuous CI/CD of ML/DL models to software-intensive components in embedded systems. In this context, we present three parallel and concurrent high-level activities that take place in companies as part of their ML/DL model development, de-

ployment and evolution i.e. Business case experimentation, Data experimentation and Model experimentation. These three high-level activities are important because companies employ ML/DL models if they bring value to the customer business, which can only be realized when data is fed as input to train models and placed into operation. We also discuss different decision checkpoints helpful for early rejection of business cases with less value. Figure 2 shows an overview of the framework and different iterations and triggers that can optimize the framework. Finally, we show how our framework help in resolving the key challenges we identified in Section 4.2. Below we detail the framework.

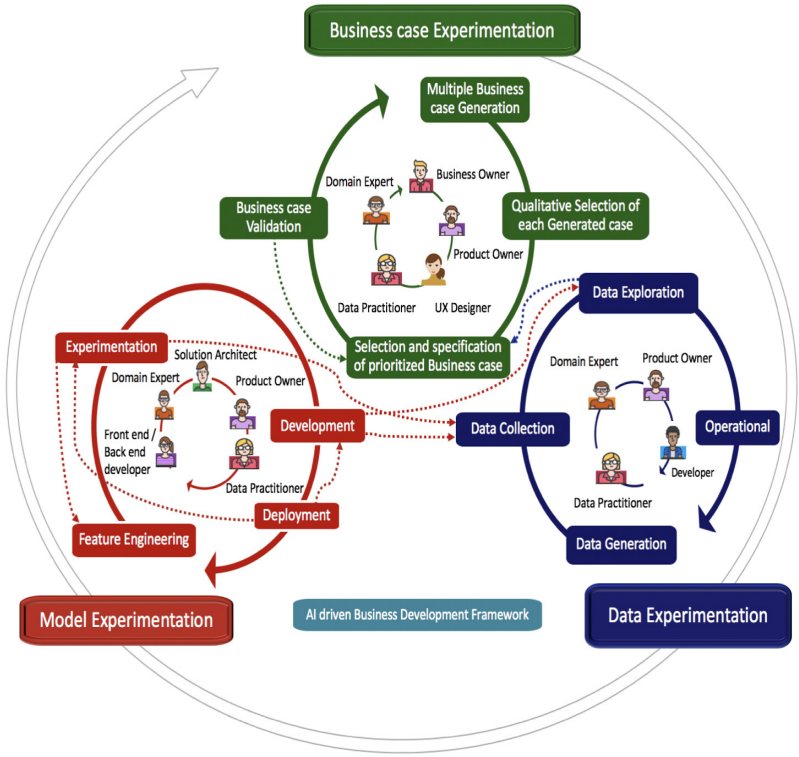

Figure 6: AI-driven Business Development Framework

5.4.1 AI-driven Business Development Framework

Our study shows that companies struggle with introducing ML/DL components, model development and the practices associated with this, into the business context of the company. In our interviews, as well as during

workshops, this was reflected by several company representatives when reporting on challenges with applying new technologies, adopting new ways of working and difficulties in introducing and educating different organizational roles in what ML/DL model development is about. In similar, challenges were found outside the direct context of ML/DL model development as these, in the end, have to be integrated and incorporated into a larger system and the overall context of a business. The end-to-end ML/DL process from business case generation to deployment depicted in the framework can be used as a blueprint for those working on ML/DL model development, deployment and evolution. AI companies will benefit only once their products are deployed into production. In such a context, this framework can accelerate the entire process and reduce the time lag between prototype and production-quality implementation of models. Also, the framework can be used as a beginner guide for people with little to no data science background and others who want to learn more about ML/DL models. Based on our learnings, we see that there are three activities that all six case companies perform in parallel and concurrently when developing ML/DL models, i.e. business case experimentation, data experimentation and model experimentation. By capturing these high-level activities and by detailing these with the roles and the iterations that take place, our framework provides a blueprint for life cycle management of ML/DL model development reflecting the continuous practices of DevOps, DataOps and MLOps. The high-level activities depicted in the framework are detailed below:

Business case Experimentation:
Business case experimentation refers to the generation and validation of business cases suitable to ML/DL. Multiple business case generation is either based on business owner needs or research projects. Often, business cases are generated as a result of meetings or ideation process or as part of brainstorming within the company. For the qualitative selection of each generated business case, companies set up frequent meetings with business owners for a better understanding of business cases and to investigate legal potential implications. A data treatment agreement can be established to ensure proper utilization of data as agreed with two parties within the scope of GDPR (General Data Protection Regulation) especially in the case of critical projects. The practitioners try to solve the business case in a non-ML/DL approach to check whether it makes sense in the beginning.

Before selecting and specifying prioritized business case, data scientists try proof-of-concept based on random algorithms or literature review to justify actual model development. Data scientists schedule early discussions on evaluation metrics with business owners and finalize the metric that has to be optimized for the business case. When the business case has been validated by business owners, it should be placed in production to make realistic data-based decisions.

Data Experimentation:
Data experimentation refers to the generation, collection and exploration of data as well as monitoring model performance based on inference data. Once the business case is specified, the next goal is to generate a dataset suitable for ML/DL models. It is shocking that the whole world praises the large availability of data and talks about the need for huge data storage mechanisms but the volume of useful data is limited in practice. During data collection, it is recommended that an early set of assumptions, questionnaires, checklists and facts to be verified about the collected data should be formulated to guide the approach and to be able to carry out an investigation subsequently. In the early stages of data collection, conduct a quality test on a sample dataset quickly as possible for evaluation. For getting a clear understanding of data during exploration, data scientists continually build and test hypotheses around the data and provide insight into non-functional requirements. Various visualization techniques are used to understand the data distributions. It is recommended to ensure that the same data pre-processing techniques are used before training and after putting the model into operation. In the operational phase, monitor the model performance. In Figure 2, operational is part of data experimentation activity since monitoring model performance is closely related to the input and output data.

Model Experimentation:
Model experimentation refers to selecting appropriate features, experimenting with algorithms, finalizing suitable algorithm and putting the model into production. Once suitable data for ML/DL models are generated and explored, relevant features are selected for modelling. The feature set is fed as input and experimented with several algorithms to find a specific/specific set of algorithms. The finalized algorithm undergoes hyperparameter tuning to optimize model performance and must

be put into production in parallel to the existing processes and systems in software-intensive companies to yield benefits. The deployed model should be placed under close supervision of A/B testing. Based on model drifts and data drifts, the model needs to be redeployed/retrained.

5.4.2 Roles and Organizational Functions

Our empirical findings highlight the involvement of different profiles of practitioners when developing, deploying and evolving ML/DL models. Once the business case is finalized, companies set up data science teams. A typical large data science team consists of product owner, data scientists, domain experts, business owner, back-end or front-end software developers, etc. All roles in the company may concurrently and parallelly involved in different ML/DL projects at the same time. Product owner in the company keeps track of best practices and all responsibilities assigned to teams, organizes work and host team meetings. Within companies, teams dealing with ML/DL projects is often split off into two: a) The main data science team that mostly works on projects with business owners b) The research team that tries the latest AI techniques for real-world business cases. All ML/DL projects demand developers to generate data suitable for specific business cases. However, it is still hard for data scientists to interact with developers who do not have a data science background. Often data scientists ask for collaboration with domain experts where they are tasked with data collection and data scientists assist them. They assist either by contributing to a tool that can be used to gather data or by discussing different ways to help them collect needed data. In case of a shortage of domain experts, data scientists can use clustering techniques or semi-supervised/unsupervised techniques for labeling.

Most companies recruit data scientists by setting high standards to save the time required to train them. Assigning a junior data scientist to work with an experienced data scientist boosts productivity and allows both parties to mutually learn and exchange skills in the companies. The company expect data scientists working in a specific business unit to quickly understand the business context and become a domain expert as the number of data scientists and domain experts is limited in companies. As a result, it is beneficial to teach software engineers how to use data science in a good, solid, repeatable and predictable manner. To accomplish this, software engineers have to be provided with different ways of working or processes

to follow or methods to pursue to deal with ML/DL models. On the other hand, engineering teams in companies try to improve the tooling and build the platform support for the project. Solution architects plan for integration of ML/DL models with the rest of the software-intensive embedded systems. Some companies even utilize interns to a) Bridge the communication gap between data scientists who deal with developing ML/DL models and solution architects who deal with deployment b) Carry out labeling tasks c) Involved in data collection d) Look into state-of-the-art learning. Data scientists retrain/redeploy new model when performance degrades and push that model to the registry of available models for A/B experimentation. Automating the business-driven development can reduce bias that occurs among data scientists while performing feature engineering for ML models and selecting favourite algorithms.

5.4.3 Decision Checkpoints for Business case Termination

Based on the analysis of our empirical data, we identify different checkpoints for rejecting business case during ML/DL model development. Below we detail various checkpoints.

The business case generated based on the needs of business owners will be terminated quickly if it makes no sense or value at the beginning stage of business case experimentation. Data scientists may treat the business case initially as a non-ML/DL case and apply basic thinking skills to verify if it creates value for business owners. On the other hand, business cases generated as a result of internal meetings or brainstorming process within the company will be terminated when the innovation management is unable to find business value for that particular case. In addition, there can be a situation where the business case needs to be terminated if the innovation management existing in the company consider the business case as valuable, but fails to find a business owner who sees value in the case. Business cases will be terminated or delayed in the absence of an existing dataset as some of the companies place a strict restriction on dataset availability. On other hand, even if the business case has sufficient data and appears fair to make predictions, the business case will be terminated if the proof of concept never justifies model development. Expensive and difficult data generation and data collection methods also lead to a stage of dissatisfaction, slowness and even termination of business cases. Business case terminates when the implementation of ML/DL solution fails as it

is not cost-effective or not approved by business owners during business case validation.

5.4.4 Iterations and Triggers

Based on empirical findings, we identified multiple iterations among high-level activities of the AI-driven Business Development Framework as illustrated in Figure 2. Although following the framework helps to establish AI-driven business development, leveraging the iterations can help in building high performing ML/DL models by ensuring better prediction, efficient inference and high business value. Below, we outline high-level activities, events that trigger these iterations, solutions to optimize the iterations as well as indicate whether evaluation metrics, data, features or model itself changes when these iterations are triggered. Figure 3 illustrates the iterations.

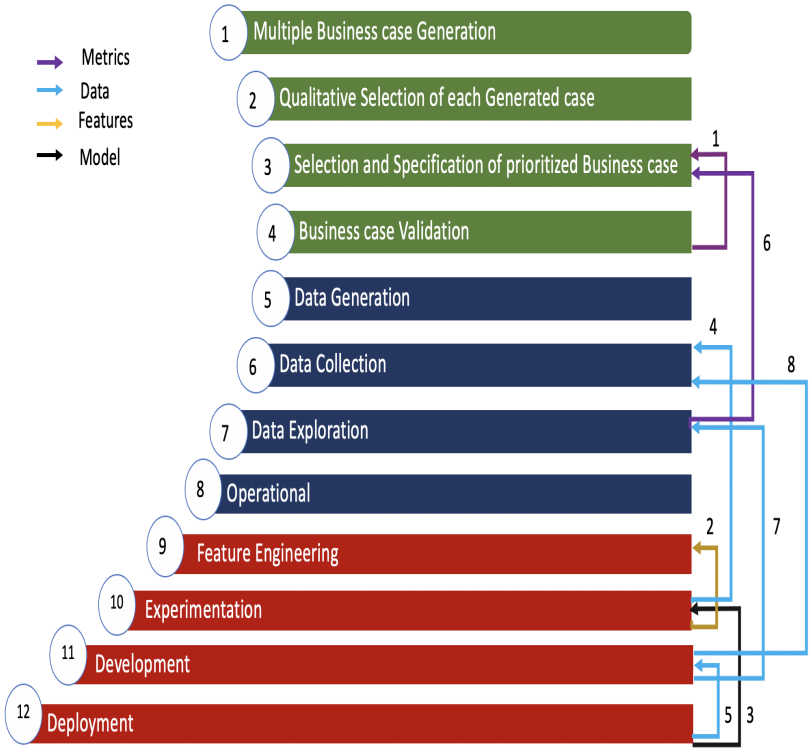

Figure 7: Iterations to optimize Design Model

1) *Business case Validation to Selection and specification of prioritized business case*:

Iteration is from business case validation to the selection and specification of the business case. In exceptional situations, business owners propose new/additional evaluation metrics during the meeting scheduled for business case validation of already agreed and defined metrics. These new/additional metrics are never conveyed to data scientists at the beginning stage of the project. This leads to a high expense and time loss. As a result, this iteration is triggered to specify the already existing business case with new metrics.

2) *Experimentation to Feature Engineering*:

The iteration is from experimentation to feature engineering. After experimenting with a simple baseline model, data scientists look into the results and try to understand the relevance of features and the correlation between each feature. This iteration is useful if it is inexpensive to get hold of features. As a result, this iteration is triggered to get a proper understanding of the system and selection of features. This triggering can provide further insights in the case of complicated data science problems.

3) *Deployment to Experimentation phase*:

The iteration is from deployment to the experimentation phase. Once the model is put into production, it makes real data predictions. We find that data scientists are looking at state-of-the-art learning to find better learning algorithms as a replacement for a well-performing deployed model. Replacement takes place when the domain or requirements changes over time. For instance, to optimize prediction accuracy, hardware cost, explainability, latency, prediction time, etc. This iteration will only be triggered if data scientists confirm that new replacement algorithms perform even better than the old deployed model and increase business value. In such cases, the model must be trained using a new learning algorithm on already existing data and features and finally proceed with the redeployment of the new model.

4) *Experimentation to Data Collection*:

The iteration is from experimentation to data collection. Data scientists validate the discrepancies between two datasets if the algorithm underper-

forms when experimenting with algorithms published in prior works. If discrepancies exist, the data collection process/collected data can be verified to ensure data quality. This iteration is triggered when it is believed that dataset verification and correction if necessary will improve model performance than experimenting with a new algorithm.

5) *Deployment to Development*:

The iteration is from deployment to the development stage. We notice that deploying the model to production is not a one-time activity in most software-intensive embedded companies. Rather, it is a continuous process. The model makes accurate predictions if the data used for prediction and training have a similar data distribution. To mitigate data drifts or to keep the model up to date, this iteration is triggered. The simplest solution is to retrain the model with new data and validate the model to ensure that model still provides accurate results. The learning algorithm and hyperparameter space remain the same as the trained model when retraining with new data.

6) *Data Exploration to Selection and specification of prioritized business case* :

Iteration is from data exploration to business case specification. Until data exploration, it is usually difficult to define the proper business goal for the project. This is because the expectation of business owners during the discussions do not equate to what can be achieved with their data. This is due to the knowledge gap between people who have a background in data science and those who do not have that background. As a result, this iteration is triggered to fine-tune the business case or even leads to business case rejection after data exploration.

7) *Development to Data Exploration*:

The iteration is from the development to the data exploration stage. This iteration is triggered when it is presumed that there are opportunities to improve model performance by revisiting and understanding the portion of data on which the model underperforms. If any interesting aspects are coined after exploring datasets, then extract the features and add them to the already existing feature set, build and train the model from scratch and proceed to deployment.

8) *Development to Data Collection*:

The iteration is from development to data collection. While trying to fine-tune the experimented algorithm that provides good results, this iteration is triggered by thinking that introducing more variances to the dataset or refining annotations will optimize model performance. In this iteration, the model is built and trained using the updated dataset and features. Once the model has been finalized, it will proceed to deployment and later put into operation for monitoring and logging.

5.4.5 Correlation between Framework and Challenges

As shown in the sections above, there are several challenges that practitioners face when developing, deploying and evolving ML/DL models. Based on the generalizations of practices and experiences of practitioners in the case companies, we developed a framework to advance their ML/DL development and deployment practices. By doing this, we seek to also help mitigate the challenges they face. Below, we describe how the framework help in resolving the challenges we identified.

Data scientists can work with developers or assist in implementing necessary tooling in the companies to generate data suitable for ML/DL business cases. Otherwise, they can ensure the dataset availability from customers themselves before starting the business case. The following iteration from development to data collection in section 5.4 provides provision to revisit the dataset to introduce variance or refine labeling to turn the dataset into more valuable and representative. If getting hold of features is inexpensive, data scientists can experiment on the selected feature set and based on results look into a larger feature set with better understanding. Because data scientists work concurrently and parallel with data science teams, they can acquire knowledge from getting engaged in different ML/DL projects. They can also collaborate with domain experts to understand the business case domain and to label the dataset. Scheduling stand-up meetings, sprint meetings, attending demos, internal workshops, etc. can make sure that every step in the development, deployment and evolution of models concerning a specific business case is known to all team members in the companies. This can ensure the intelligibility of models and avoid risk when data scientists are assigned to different projects in the middle of working on a particular project. Working together with various expertise in different projects can reduce misconceptions about ML/DL models

among data scientists. For instance, DL models are complex compared to ML models. Adopting DL models can reduce the significant time needed for feature engineering. To avoid training-serving skew always make sure that both training and test data undergo the same process of data collection, cleaning, etc. and have the same formats. Also, retraining models with more data in appropriate format can reduce training-serving skew to an extent. When datasets are collected, make sure that even corner cases are captured so that we can reduce this skew. When model drifts occur, utilise the experimentation stage and experiment with a better algorithm if drift is due to a change in technology or domain. Performing proof-of-concept can reduce time spend on model development and can achieve a realistic idea of what can be achieved with the business case. Based on proof-of-concept, data scientists and domain experts can be allocated to projects with high value. Since business owners are also involved in the business case experimentation, communication with business owners can be improved to a greater extent by asking them to fill up questionnaires, attend frequent meetings, educating them, etc.

5.5 Related Work

5.5.1 Developing ML/DL models and Associated challenges

ML/DL systems vary from traditional software in the following ways: a) Since ML/DL models learn from data, collecting, processing and updating data takes time b) In addition to SE skills, teams need in-depth ML/DL knowledge to develop high-performing models c) Opposed to traditional software development, model deployment requires extra effort. Furthermore, the mechanisms for monitoring and logging are unique to ML/DL models. According to reference [21], experts, intermediate users and amateurs are involved in developing ML models. Amateurs are non-experts who are unacquainted with ML technology. There has been very little research conducted on the work of intermediate users and amateurs. According to reference [13], expert programmers experienced difficulty in applying ML. Reference [109] investigated implementation of ML systems by software engineers. Various tools are available to support people involved in ML/DL technologies. While ML/DL offers powerful tools, the knowledge needed to apply these tools to specific scenarios remains a challenge. Smaller non-profit companies have extensive local and domain expertise but lack the skills to implement ML/DL to their context [29].

Reference [22] mentioned current working practices of data scientists and the impact of AutoAI (Automated AI) on these practices. Reference [28] proposed data processing approaches adopted by data scientists. A holistic process model that explains activities, initiation, error estimation and deployment of a Supervised ML classification model is presented in [128]. The ML workflow [18] has nine stages, including data-oriented and model-oriented stages. A high-level overview of the ML process used for developing intelligent systems is described [13]. Data management, model learning, model verification and model deployment are the activities associated with four stages of the ML workflow as defined [111]. These are similar to data science workflows such as CRISP-DM (CRoss-Industry Standard Process) [58] and KDD (Knowledge Discovery Databases) [57]. CRISP-DM enables companies to apply data mining in real-world scenarios regardless of technology. KDD is used to discover knowledge from data by using data mining tasks. Both of these workflows have a data-centric focus with several feedback loops between stages.

People working on ML/DL model development, deployment and evolution encounter several challenges. Reference [10] attempts to identify and classify numerous SE challenges that arise when developing and deploying ML components in software-intensive systems. Experts encounter issues when developing systems using ML algorithms, as described [13] [14]. Reference [21] investigated how non-experts develop ML model based on their experience, knowledge and blind spots and also revealed their unique potentials and pitfalls. ML specific risks that should be considered during system design using the SE framework of technical debt is illustrated [15]. Some of the technical debts are boundary erosion, hidden feedback loops, data dependencies, configuration debt, changes in the external world, etc. Studies have been performed to test software [59] and ML models. Challenges at the intersection of SE and ML has been described [9]. However, studies considering the combination of SE and ML have not been extensively researched [60]. Similar to software, implementation of a production-ready ML system needs testing [19] and can be grouped into development, production, and company challenges. In contrast to ML and traditional software development, DL technology needs specialized infrastructure support [61] as scaling up DL models boosts performance to a greater extent [61].

5.6 Threats to Validity

We concentrate on mitigating validity threats by focusing on construct validity, reliability and external validity [62]. To enhance the construct validity of our research study, the authors and involved practitioners are most familiar with developing, deploying and evolving ML/DL models. We employed multiple techniques (interviews, workshops, meetings and events) and sources (senior data scientist, technology specialist, AI application specialist, etc.) for collecting empirical data. Our study involved six case companies and the results were reviewed by practitioners who were both, directly and indirectly, involved in our research. To ensure reliability, we validated the findings with practitioners at the companies during workshops, meetings and events. In the case of external validity, the key contribution of the study can be applied to similar software-intensive embedded systems companies interested in developing, deploying and evolving ML/DL models. The reported empirical findings are based on interviews as well as insights and observations from workshops, meetings and events where the practitioners share their subjective perceptions and experiences.

5.7 Conclusion

Though ML/DL technologies are gaining more popularity, companies face several challenges when designing ML/DL models. To address these challenges, and to help embedded systems companies in advancing their ML/DL model development and deployment process, this research study focuses on designing an end-to-end process of developing, deploying and successfully evolving ML/DL models. In this study, we identify high-level activities that companies perform in parallel and concurrently to develop, deploy and evolve models. Furthermore, we detail the activities, iterations, and triggers that optimize model design as well as roles and organizational functions. We also show how this study helps companies in resolving challenges that we identify and discuss different decision points for immediate termination of less value business cases. In future research, we plan to focus on the continuous delivery of MLOps as well as how the adoption of MLOps impacts existing practices in companies and the associated challenges.

6 AI on the Edge: Architectural Alternatives

This chapter has earlier been published as
M. M. John, H. H. Olsson, and J. Bosch, "AI on the Edge: Architectural Alternatives," In 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), pp. 21–28, IEEE,2020

Current IoT solutions rely more on a cloud-based centralized model for (re)training of ML/DL (Machine Learning/Deep Learning) solutions as it offers high-performance, centralized computing and unlimited storage capacity. Data transfer to cloud leads to issues related to processing power and latency [129] [130]. Shifting computing capability from centralized cloud to distributed edge nodes solves the aforementioned issues and enables low latency and reliable intelligent services. It brings two major enhancements [131] over existing cloud infrastructure: a) Most data pre-processing can be performed on the edge before being sent to cloud ii) Edge with computing capabilities can optimize cloud resources [132]. Nevertheless, edge nodes need to reduce the cost and power demanded by ML/DL (re)training to be distributed across the edge and cloud [131].

Most software-intensive embedded systems companies are currently relying on a centralized cloud approach and are in the midst of a transition to enable computing capability at the edge [133] [134] [135]. When companies move from cloud-to-edge, they face difficulties in deciding the amount of information to be collected and transferred to the cloud, handling labeling challenges and controlling their quality, installing cloud and edge model versions, implementing general platform to easily update algorithms when a new model performs better, reporting metrics back to the cloud,

monitoring solutions and managing logging mechanisms. There is, therefore, a need for guidance in terms of what architectural alternatives exist and how these allow companies to improve their edge/cloud (re)training activities.

Based on action research in which we study multiple use cases in a software-intensive embedded systems company, as well as insights from long-term collaboration with companies in the embedded systems domain, we develop a generic framework for deploying AI on the edge utilizing transfer learning. We validate the framework in a qualitative interview study with four additional case companies and present the challenges they face while selecting the ideal architecture. The contribution is threefold. First, we develop a generic framework consisting of five architectural alternatives, ranging from centralized cloud architecture to decentralized edge architecture. Second, we validate the framework in a qualitative interview study with four additional companies and as an additional result, we present two variants to the architectures identified as part of the framework. Finally, we identify the challenges that experts face when selecting the ideal architectural alternative.

The rest of the paper is organized into five sections. Section 2 sets out the background. Section 3 introduces the adopted research methodology as well as the case companies involved in our research study. Section 4 presents the generic framework consisting of five architectural alternatives. Section 5 discusses validation study findings and points out the challenges associated with the selection of each architecture. Finally, we conclude the paper and discuss future research in section 6.

6.1 Background

6.1.1 Edge/Cloud (Re)Training

Cloud computing facilitates the training and inference process for typical ML applications with the support of centralized data centers with high capacity for large scale data storage and processing [136]. When edge devices generate massive amounts of data ranging from sensors to robots, computing services undergo a cloud-to-edge transition. Companies often see this transition as an opportunity to minimize costs and to strengthen privacy and low latency, as most of the processing, storage, and networking resources are located in powerful data centers owned by Google, Amazon,

Microsoft, Facebook, and Apple [137]. According to [138], edge computing and ML are two sides of a coin. Perform model retraining carefully as data patterns change frequently without warning which makes ML/DL model obsolete in a short time. Based on the amount and path length of data offloading in [139], edge intelligence rates at six levels where training and inference is entirely cloud-based and gradually focuses on edge going forward. Studies in [134] [140] discuss state of the art, key characteristics, requirements, design patterns, quality attributes and architecture modules for addressing big data (reference) architectures.

6.1.2 Transfer Learning

In transfer Learning [133] [141], a generic model is predominantly trained in the cloud and the resulting model is optimized with incremental training at each edge node at local sites. It is an ideal technique to boost model performance when training data is costly or difficult to collect, labeling tasks are expensive and to reduce infrastructure costs and training time. The [142] offers an overview of transfer learning, its applications, extensive details on specific transfer learning solutions, such as, homogeneous, heterogeneous and negative transfer learning. Generic model retraining increases accuracy and updates must be sent to the edge instances on a regular basis for better performance.

6.2 Research Method

The research we report in this paper is part of a larger research initiative in which we conduct longitudinal multi-case study research in close collaboration with fifteen companies in the embedded systems domain[2]. The collaboration includes a broad portfolio of research projects and within this context, we have been working with a number of companies in the area of ML/DL. For the purpose of this paper, we present findings based on multiple use cases in one of the companies. However, it should be noted that our insights are a result of a long-term collaboration with a number of companies involved in the research initiative. Based on these insights and the specific learnings generated when studying different ML/DL use cases in a software-intensive embedded systems packaging company, we develop a generic framework consisting of five different architectural alternatives to deploy AI on the edge. To validate this framework, we conducted a qual-

[2]Software Centre

itative interview study with selected experts in four additional companies that were not part of our previous or on-going research on this topic.

6.2.1 Action Research

Action research [143] was conducted to understand edge/cloud (re)training, inference and transfer learning in the context of ML/DL use cases in the packaging company. It is a research approach in which researchers work in close collaboration with experts as partners in the process of knowledge creation in the context of practice. The researcher has to get into a company and engage in experiment of learning with experts. It can be used to address complex real life problems and immediate concerns of experts. It appears to be a well suited research method for this study due to its emphasis on collaboration between researchers and experts in real-world settings.

6.2.2 Case Company

In this section, we present the case company where we conducted the action research study and we detail the four different ML/DL use cases (A1, A2, A3 and A4) that we studied.

Packaging Company: The company provides packaging and processing solutions for food products. They also offer packaging machines to different customers. The company uses ML/DL models to ensure customer convenience, product quality improvement and cost reduction. A camera has been installed at the packaging production line to maintain package quality and whenever quality degrades, an alarm has been generated to stop production to reduce costs.

A1 - Defect Detection: The company detects defect in fully finished packages at each customer site. Global model at cloud is deployed at each customer site for inference through transfer learning to detect customer-specific defects more effectively. Learnings from each customer is send back to the cloud for retraining. The dataset consists of fully finished packages with different patterns and types.

A2 - Quality check of Sealing: The company checks inner side of the packages to detect flaws or deviations in the sealing. Temperature, anomalies and package edges are analysed to ensure quality of sealing. It undergoes various tests to determine whether they are underheated or overheated as it becomes impossible to connect the internal part. The dataset consists of packages with different patterns and types.

A3 - Anomaly Detection: The company detects anomaly in semi-finished packages. The package will be recreated and proceeded to production depending on whether it is flawless or not. Detecting defect in a semi-finished package can reduce effort, time and cost as their recreation can be stopped at an early stage. The dataset consists of semi-finished packages with different patterns and types.
A4 - Food Misplacement Detection: The company provides multiple filling options for food products. They detect defects in placement, position, alignment etc. of food products to meet evolving customer demands. The dataset consists of food products with different shapes and sizes.

6.2.3 Data Collection and Analysis

The first author spent two days a week on-site as part of action research and work with the data scientist team for a period of six months from September 2019 - March 2020. Experts involved in action research are shown in Table 1 where *P** denotes action study participants from the packaging company. By being present at the company and actively involved in everyday development and practice, the first author gained an in-depth understanding of the case company, its environment, employed technologies and challenges experienced by the team when working with the use cases. Based on this understanding, and by working closely with experts in the field, a generic framework was developed by the first author for deploying AI on the edge. The general process cycle of action research used in this study involves diagonizing, action planning, action taking, evaluating and specifying learning [144]. After frequent discussions with experts from the company, we identified a particular problem, collected relevant data as part of the action planning to develop a generic framework consisting of five architectural alternatives, evaluated the framework with experts and incorporated their feedback. The action research was complemented by working closely with the team of experts in the packaging company and participating in their project meetings. In addition to this, authors frequently conducted several interviews, meetings, discussions and workshops with experts (including one-to-one and group meetings) to collect data. The data collection was further supported by taking meeting notes, writing diaries and through presentations given by the company. Analyzed the collected data by creating a summary and discussed it with company experts via presentations and also communicated to the other two authors.

Table 7: Description of experts involved in action research

ID	*Roles*
P1	Data Science Head
P2	Data Science manager
P3	Senior Data Scientist
P4	Director - Industrial IoT System
P5	Manager - Data Processing and functions
P6	Solution architect
P7	Software Engineer
P8	Technology Specialist
P9	Data Scientist
P10	Data Scientist
P11	Data Scientist

The architectural alternatives developed by the first author based on the action research was presented to the experts in the packaging company and incorporated their inputs and concerns.

6.2.4 Validation Study

To validate the generic framework, we conducted qualitative interview study with selected experts in four additional case companies. None of these experts had been involved in our previous research on this topic and therefore, could provide valuable feedback in relation to the applicability and/or the challenges they identified with each of the identified architectural alternatives.

Case Company W - Automobile: A joint engineering and development centre for passenger cars that address the needs of two large companies. The company utilizes ML/DL models to provide better mobility solutions to the customer.

Case Company X - Telecommunications: A large telecommunication company enables intelligent network operations to solve problems faced by service providers. They introduce ML/DL techniques to increase both revenue and customer satisfaction.

Case Company Y - Pump supply: The company develops different kinds of pumbs, electric motors and electronics for pumb control. The company

employs ML/DL models to collect mechanical and operational data to provide actionable insights on health of machines and diagnostics over products.
Case Company Z - Manufacturing: The company employs ML/DL models to analyze data collected from different products ranging from home appliances to vehicles out in the field to enhance design, functionality and quality of products.

The qualitative interview study [83] was conducted between February to March 2020 following an interview guideline. Part I of the interview guideline focused on the role and experience of the interviewee. Part II focused on the validation of developed generic model and current architectural alternatives that exist in each case company. Finally, Part III explored the challenges faced by experts while selecting an ideal architecture. All interviews lasted one hour and were conducted via video conferencing. The interviews were recorded and transcribed for analysis with the consent of the interviewees. The experts involved in this study are summarized in Table 2 where V * denotes validation interview participants. The interview summary was cross-checked with audio recordings, interview transcripts and notes taken during interviews. We analyzed the interview summary to identify key concepts and then grouped them into conceptual categories. We used triangulation to boost precision and enhance research study validity [98], which relies primarily on qualitative data. Since we were three authors conducting interviews and reading transcripts, investigator triangulation was used to take multiple perspectives on the topic under study, thereby providing a broader picture. The goal of the first iteration was to validate the developed generic framework. The second iteration was carried out to present two variants to the architectural alternatives identified as part of the framework and to identify the challenges faced by experts when selecting architectures.

6.2.5 Threats to Validity

We try to mitigate the validity threats related to our work, particularly construct validity, reliability and external validity [145]. To strengthen the construct validity of our research, authors and experts involved in the study are most familiar with edge/cloud (re)training and inference. We also used multiple techniques (action research, semi-structured interviews, meetings, discussions and workshops) and multiple sources (data science

Table 8: Description of experts involved in validation study

Case company	Experts	
	ID	***Roles***
W	V1	ML Engineer
	V2	ML/AI Engineer
	V3	Vice President
X	V4	Senior Researcher
Y	V5	Head of AI Technology
	V6	Senior Data Scientist
Z	V7	Senior Software Architect
	V8	Data Scientist
	V9	Data Scientist

head, data science manager, data scientist, ML engineer, etc.) for data collection and validation. The action research was performed for a period of six months with the participation of experts in the packaging company. In order to ensure reliability, the findings were validated with experts from four different case companies. With regard to external validity, we forsee threat as to how this research can be applied to other companies on the basis of a few use cases. We believe that by extending this research to more software-intensive embedded systems companies in the future can mitigate this threat.

6.3 Architectural Alternatives

Based on our research, we develop a generic framework of five different architectural alternatives for deploying AI on the edge utilizing transfer learning. Fig 1 shows all concepts used for developing the framework. Across all five architectural alternatives, the model at cloud, which will get deployed to edge is referred to as *global model* and the model at edge (at each local customer site) as *child model*. If additional data is provided to the existing model, it is referred to as *retraining* and, if additional/existing data is provided to the improved model, it is referred to as *training*. All architectural alternatives have a *shadow model* in addition to the global model at cloud, which is constantly (re)trained using *global dataset*. The global dataset is obtained either by direct sampling or by synthetic sam-

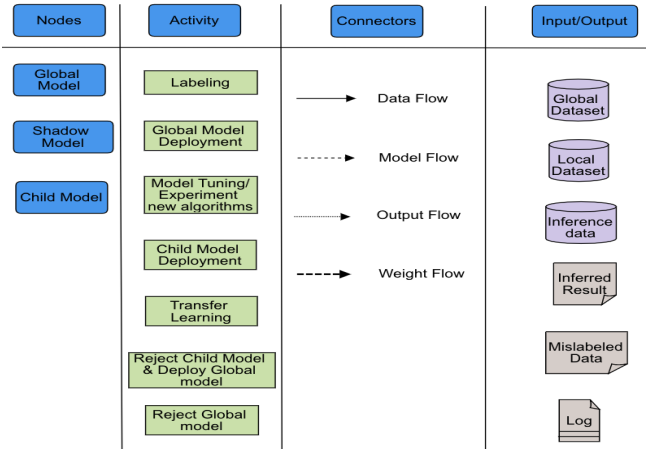

Figure 8: Concepts used for developing generic framework

ple generation within the company. Based on performance comparison between shadow model and global model, high performing shadow model is copied to global model and the global model gets deployed to the edge. The model performance can be evaluated by using different metrics such as mean average precision, accuracy, false positives, false negatives, etc. The global model can be compressed using different available compression techniques before being deployed to the local sites. The *local dataset* for child model training is composed of specific data available at each local site. Both global and local dataset undergoes *labeling* before model (re)training. At each local site, *inference data*, mostly in streaming mode predicts the child model performance.

The model management can be achieved by sending *logs* back to the global site when Type 1 or Type 2 error occurs in order to get an indication of whether or not we are in align with the results seen during validation. Log files help to keep track of all changes that have to be made for better performance. Based on basic concepts shown in Fig 1, Fig 2-6 shows an overview of the generic framework of five different architectural alternatives. In the sections below, we detail each architecture.

Architecture 1: In the first architecture, the traditional ML/DL systems adopt more of a centralized approach where global model (re)training takes place in the cloud and a child model is placed into local operation at the edge.

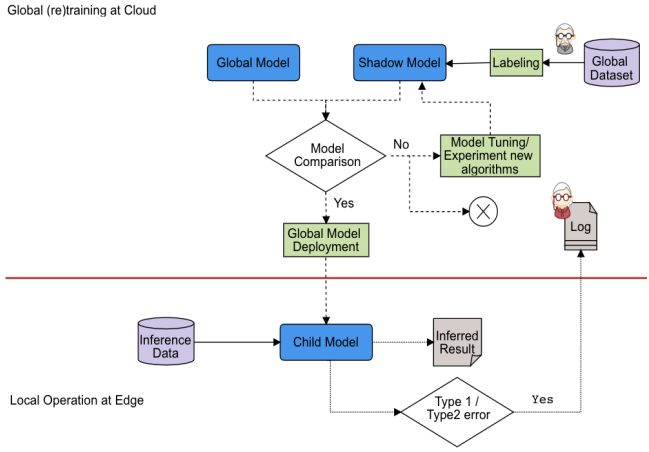

Figure 9: Architecture 1: Centralized approach

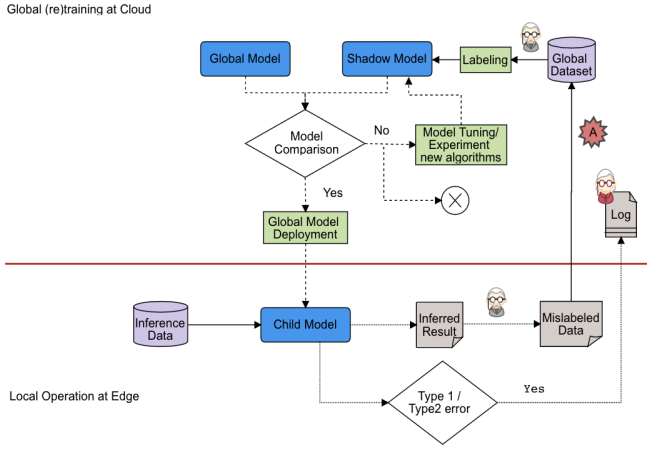

Figure 10: Architecture 2: More Centralized, Less Decentralised approach

> P2: “It is easy for big companies who own clouds or companies who have created their own HPC infrastructure to implement cloud solutions. For others, it is a bit expensive"

The dataset needed for global training requires data to be centralized on a machine or on a data center. The labeled global dataset is supplied to the training model. When training the shadow model for the first time, it is copied to the global model without any model comparison. The shadow model is constantly evolving in this architecture and when it outperforms the global model, it is replaced with the shadow model and deployed on the edge. If the shadow model under performs, the performance can be improved either by hyperparameter tuning or by trying new algorithms which best fits the business case. If the shadow model is still not performing as expected, rely on previously deployed global model without taking any further action. The global model deployed on the edge is compressed without decreasing model performance.

> P3: “When you apply compression techniques carefully, you usually loose 1 % of average precision and all the things you need to classify. It would not decrease much performance and sometimes it remains the same. It is quite robust"

The child model produces inferred result as part of inference at the edge. The log files generated from edge generally consists of statistical information related to the inference results such as correctly classified results, incorrect results, the total number of data points that the model has seen so far, the general average precision or inference accuracy at the edge for every 1000 data points, etc. Log files are sent back to the global site for monitoring. Whenever global model outperforms the shadow model at cloud, the global model is deployed towards the edge may be once in a month or once in every agile sprint.

Architecture 2: In the second architecture, the global model (re)training occurs in the cloud and local operation happens at the edge. the main difference between architecture 1 and 2 is that mislabeled data identified after inference with the help of human intervention or with the help of any unit testing is send back to the cloud for improving quality of global model (marked as “A" in the diagram). The mislabeled instances consist of two types: a) correct instances that are misclassified by the model, b) incorrect instances that the model fails to detect. Both instances are examined by the

human-in-the-loop and are added to the global dataset for model retraining. If the company has enough bandwidth, mislabeled data together with all data that has been correctly classified as defects, 20% of local data and may be 10% of local data in case of any catastrophic event happens can be send to cloud. All the mislabeled instances have to be labeled before sending for global retraining. In contrast to architecture 1, the log files contain mislabeled instances in addition to all other information. To deploy a global model at local sites and to send back needed data for global model retraining, a strong internet connection is required.

> P3: “Deploying at the edge is the easiest part. Compressing the network and writing the software to do inference on the edge is not that difficult. The challenge is how we do the retraining of global model. Should we do 24/7 retraining or allocate a virtual machine when there is a need to retrain?"

> P6: “Internet connection may really seems less important but some of our customers are in really remote locations"

Architecture 3: In the third architecture, global model (re)training occurs in the cloud and child model (re)training and placement into local operation happens at the edge. In addition to architecture 2, the global model utilizes transfer learning technique to transfer certain layers to the child model in order to boost the performance. The transfer learning applied child model is then trained on local data to suit local customer specifications. The performance of the transfer learning applied child model is then compared with the global model. If the child model performs well, deploy it to the local operation for inference. Otherwise, apply transfer learning to some more layers of the child model to improve performance. Reject the child model if it is still under performing and proceed with the deployment of global model to the local operation with the assumption that global model is being trained on data gathered from different customers. At each local site, child (re)training will be probably carried out with the help of ML/DL automated applications like AutoML as they experience lack of ML/DL experts. Larger companies can apply transfer learning on their own datasets, but for smaller companies this is not the case. The copyrights on exploited datasets act as a hindrance when downloading pre-trained model to utilize transfer learning.

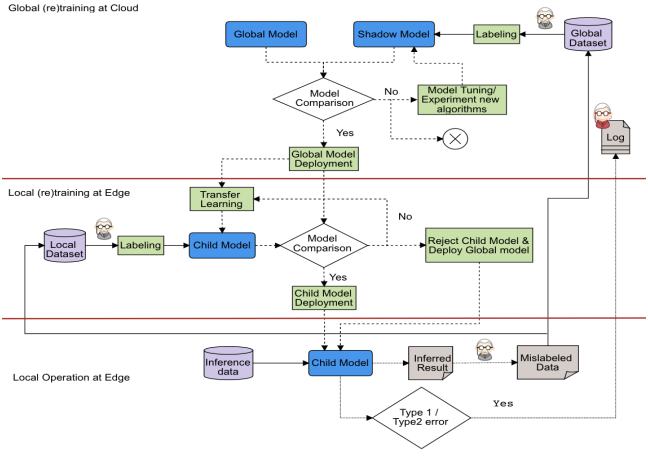

Figure 11: Architecture 3: Mix of Centralized and Decentralised approach

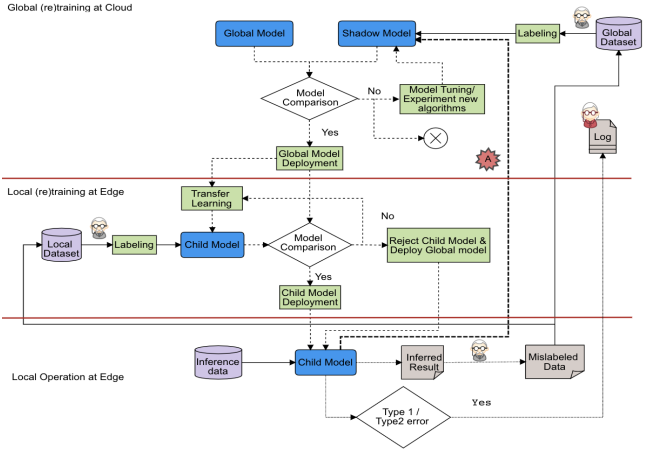

Figure 12: Architecture 4: Less Centralized, More Decentralised approach

> P3: “Exploiting pre-trained models that are already trained on millions of publicly available datasets and then applying transfer learning is not giving us good results. Most of the time the exploited dataset has copyrights, then you need to train from scratch. May be you need to train several times because a lot of parameter setting is empirical"

The misclassified instances are labeled and send to retrain the child model for better performance with the help of human-in-the-loop. The log files contain all information related to the inferred results similar to architecture 2.

Architecture 4: In this architecture, global model (re)training takes place at cloud and child model (re)training and placement into local operation happens at the edge. In addition to architecture 3, the learnings from the child model needs to be bring back to the global model for better optimization. This can be accomplished through some form of fine-tuning of the model where the global model and child model vectors may undergo operations such as element multiplication, element addition, etc. (marked as “A" in the diagram) and results in a new shadow model. The fine-tuned shadow model can be trained on the global dataset which consists of mislabeled data send after the inference from each local site to achieve more generalization. The model can be then deployed to local operation utilizing transfer learning to boost performance.

We have observed that companies would benefit if they have production focused and research focused teams working in parallel. The production focused teams can concentrate on traditional centralized approach where most of the (re)training happens in the cloud, while research focused team should try new concepts related to edge (re)training.

> P3: “Sometimes you/your team need to do standard pipeline. Some other team needs to try transfer learning, federated learning, etc. but sharing these information to the guys who already put those things in production is the real deal"

> P6: “It is too difficult to reach out to people outside the company to discuss things"

Architecture 5: In this architecture, global model (re)training occurs at the cloud and the child model (re)training and placement into local operation

happens at the edge. In addition to the process in architecture 4, there might be a situation where global model does not achieve much better performance even after retraining with mislabeled data, transfer learning and fine-tuning. If the global model does not work well as expected When compared to child model, companies may reject the global model and focus only on the child model. They can retrain child model with local data available at each local site and place them into operation for inference.

> P3: “If you don’t need much knowledge, we can reject global model"

Companies can reduce high costs related to data transport and model training at cloud by employing this architecture. The logs are frequently checked by human-in-the-loop to indicate model performance degradation.

> P6: “I don’t even actually think that big technology companies are able to safely reject the global model, but I am 100% sure that this is actually what companies are trying to achieve"

6.4 Validation Study Findings

To validate our architectural alternatives, we conducted interviews with experts in four additional software-intensive embedded system case companies that had not been part of our research on this topic. Below, and based on their input, we present the validation results for each architecture and the challenges they identified in Fig 7. As a general observation from our validation study, we see most companies rely on architecture 1 and are trying to shift from a completely centralized cloud approach to a mix of centralized and decentralised edge/cloud solutions.

Architecture 1: According to case company W, most of their business defaults to the centralized cloud-based architecture as they have many locals. V4 from case company X also considers architecture 1 as the classical way of working in telecommunication networks.

> V4: “This is the default case which today’s network has. To be frank, there is not much ML in today’s networks. Since Telecom has 6 nines, we cannot go with probabilities too. There are places where now it is getting into real network, but not much”

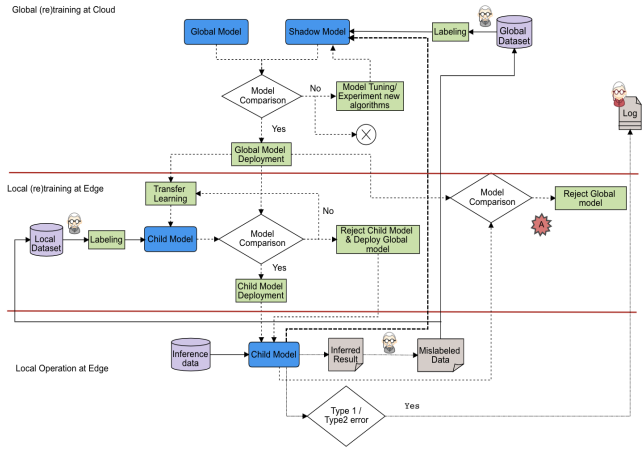

Figure 13: Architecture 5: Decentralised approach

For case company Z, most of their use cases are still relying more on a centralized cloud approach.

> V4: “Most of our use cases are still in this area, but we strive for going more for the edge"

According to case company Y, dataset needed for global training has to be set up in labs. When expected accuracy is achieved within training pipeline, they come to a conclusion that dataset collected is adequate for global training. They also think that global model compression before deploying to edge depends on available bandwidth for sending model across edges. V4 also agrees that it is better to compress and decompress global model if it is huge. We also observe that companies use logs to keep track of model life cycle.

> V4: “We need log files to find model life cycle and also to figure out how models adapt to concept drifts"

Architecture 2: According to case company W, retraining global model with mislabeled data is a good addition as outlined in architecture 2. They also consider similar architecture to retrain their global model during deployment of intrusion detection system when their local operation gets completely messed up. V7 suggests that applying transfer learning with new data can reduce the time taken for retraining.

> V4: “When retraining takes too long, you would want to do some form of transfer learning with new data"

V4 finds it very difficult to collect labeled data from edge as customers are reluctant to send data as they are either critical or valuable for them.

> V4: “The biggest problem for us is to get labeled data when it is deployed on the edge and Type 2 errors are really difficult for to capture"

Case company Y find it less feasible to send mislabeled instances for global model retraining. According to their opinion, it is expensive to send all data to the cloud as they have a lot of high frequency data getting produced at edge.

Architecture 3: According to V1, proper care should be taken while selecting the local dataset, otherwise it will lower local model performance even after applying transfer learning in architecture 3.

> V1: “It would be strange if local dataset is worse. It may produce less performing model even after doing transfer learning with local data"

We have observed that case companies employ transfer learning technique heavily in their companies. Transfer learning has been used by DL models in case company X to translate documents that are available across company such as product specification.

> V4: “Transfer learning is one space where people try neural network. We try removing the fully connected layers and play with the convolutional layers"

Case company W and Z think that there should be less human-in-the-loop for all architectural alternatives. According to case company W, they are not interested in encouraging more human in the loop thinking they are customers. They also observe that finding and sending mislabeled data to local dataset after proper labeling needs more customer involvement. They find that architecture 3 works for them in future when they have enough computational capacity at the edge.

> V1: “I would say no and the reason is that there is already human-in-the-loop to check when things messed up at edge and to send mislabeled instances for global model retraining. Human in the loop won’t be successful for us because they are our customers and we don’t want them to do lot of stuff...At the moment, we don’t have enough computational capacity at the edge to be able to do this”

> V8: “Automation will be very important and highly prioritized. Human-in-the-loop is something that we want to avoid in our cases”

Architecture 4: Case company W consider that sending learnings back to the global model is the most difficult and important part in architecture 4 and also observe that if not careful, it may end up with a local model representation at the global level. They also think that it is good to have more local optimization for a specific problem.

> V1: “I think the tricky part is how to send learnings back to improve the global model and to get the benefit out of that....Local model has much more narrow distribution. If you are training it for long time, optimizing it and sending those parameters back to the model, you have to be pretty careful of what you do, so that you don’t get a representation of a local model. So that is an important step”

> V3: “It is nice to have more local optimization for a specific problem"

According to case company Z, architecture 4 seems to be an entirely new concept and very interesting which they would like to apply in their context.

> V7: “Architecture 4 is very interesting to look at, but it does not resemble what we have done so far”

Architecture 5: Most of the case companies consider architecture 5 as expensive and show less confident on rejecting the global model and become stand alone.

> V7: “Architecture 5 seems to be expensive”

Case company Z expect that the global model cannot be rejected if the variance among child model is high and can be rejected if the variance is limited. Companies find both strength and weakness in architecture 5 as there is a lot of local operation and (re)training can be done at edge which helps in cost reduction by rejecting the global model. At the same time, companies find it difficult to reject global model if the child models vary a lot from each other.

> V8: "Global model can still be relevant when child models deviate in different directions"

Additional variants of Architectural Alternatives: As part of our validation study, we got input describing two variants to the architectures we identified as part of the framework. The first variant is *quorum of child models*. In absence of a global model, child models in a certain region or domain integrate their respective models and learnings by itself. Hence, in between the global and the local layer, there will be a subset of child models benefiting from each other. The second variant is *wrapper around global model*. In this variant, a wrapper is placed around the global model so that you can do adjustments outside the wrapper while you still deploy the global model continuously and with protected performance.

From the perspective of experts involved in validation study, most of them consider architectural alternatives 1 to 3 as traditional way of working and architectures 4 to 5 as more related to edge learning. In viewpoint of V4, architectures 1 and 4 are more familiar to the companies. Some experts think that architectures 3 to 5 will be potentially relevant in future when they have relevant capacity to do local training.

6.5 Conclusion

Deploying AI on the edge is gaining increasing interest in embedded systems domain companies, but experts face challenges in deciding the ideal architectural alternative. In this paper, based on action research we develop a generic framework consisting of five architectural alternatives for the effective implementation of edge/cloud (re)training. We validate our findings with experts from four case companies. Based on the findings of the validation study, we present two variants of the architectures and identify the main challenges that experts face when deploying AI at the edge. We believe that the results can provide valuable insights to companies

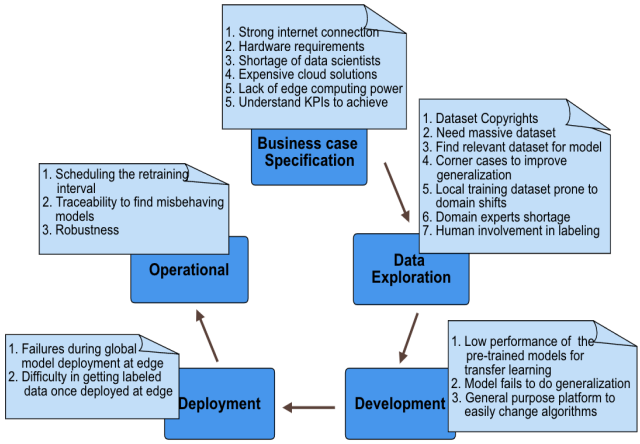

Figure 14: Challenges associated when determining each architecture

that are in the midst of implementing edge/cloud (re)training solutions. In our future research, we plan to expand our study by involving additional case companies and use cases and to further validate our results with experts in the field. Also we look to further explore factors, prioritization among factors and trade offs to be considered when experts decide on what architecture to implement in practice.

7 AI Deployment Architecture: Multi-Case Study for Key Factor Identification

This chapter has earlier been published as
M. M. John, H. H. Olsson, and J. Bosch, "AI Deployment Architecture: Multi-Case Study for Key Factor Identification," In 27th Asia-Pacific Software Engineering Conference (APSEC), pp. 395–404, IEEE, 2020.

With the prominence of IoT and Big Data, companies are increasingly integrating AI/ML/DL (Artificial Intelligence/Machine Learning/Deep Learning) [18] [146] techniques in their systems to improve business value delivery. Also, the high performance of parallel hardware such as GPUs and FPGAs speeds up the adoption of these techniques in companies [2]. ML/DL models have been successfully developed and deployed in a significant range of companies, including healthcare, financial services, automotive, telecommunications, retail, manufacturing, defence, etc.

Even though they are highly adopted in companies to provide actionable insights, most companies find the transition from prototype to production quality deployment models as quite challenging [9] [10] [11]. Companies face significant challenges in monitoring and logging, model testing, resource limitations, troubleshooting, data sources and distribution, glue code and support, privacy and data safety, data silos and data storage when deploying ML/DL models [2]. Apart from these, one of the critical challenges that practitioners face is how to determine the selection of an optimal architecture for AI deployment. There exists a lack of well-established guidelines for the development and maintenance of ML/DL models in companies [147]. In addition to this, model deployment is a

highly underestimated area [2]. So, we believe that guiding how to select the optimal architecture for a certain AI deployment can potentially accelerate the deployment process in companies.

Based on our previous research [148], we developed a framework in which we identified five architectural alternatives for deploying AI ranging from centralized cloud to fully decentralized edge architectures. We validated the framework in four software-intensive embedded system companies and identified various challenges they face when deploying ML/DL models. In this paper, and to advance our research, we identify the key factors to be considered when selecting the optimal deployment architecture. For this, we conducted a follow-up study involving interviews and workshop in seven case companies in the embedded systems domain. Based on these findings, we develop an architectural selection framework in which we outline how prioritization and trade-offs between these results in a certain architecture.

The contribution of the paper is threefold. First, we identify key factors that are critical for AI system deployment. Second, we present an architectural selection framework in which we outline how prioritization and trade-offs between key factors result in the selection of a certain architecture. Third, we discuss additional factors that may or may not influence the selection of an optimal architecture in addition to the key factors. The rest of this paper is organized into six sections. Section II describes the background. In section III, we introduce the research methodology adopted for conducting the study. Section IV focuses on the findings of the study. In section V, we present the architecture selection framework and present discussions in section VI. We detail threats to validity in VII. Finally, section VIII summarizes our conclusions and future works.

7.1 Background

7.1.1 AI/ML/DL

AI has been used in the fields of computer vision, robotics, the science of cognition and reasoning, natural language processing, game theory, etc. [149]. With the pace at which ML/DL technology has been introduced, life improved in real-world scenarios. Methods, recent developments and research opportunities in ML are summarized in [100] and [101] provides an overview of DL concepts and techniques. Compared to DL, a lot of

effort and time is expended on feature engineering to identify data patterns. ML/DL differs from conventional software engineering since its behaviour depends heavily on external data. The primary distinction between ML/DL and non-ML/DL systems is that data replaces code within the ML/DL system [9].

7.1.2 Model Deployment - Challenges

According to [15] and [126], developing, deploying and maintaining complex commercial ML-based system is a challenging task. Most ML-based systems have strict latency requirements at inference stage [150]. Training-serving skew also results in sub-optimal model performance [151]. For the realistic implementation of ML, there is a need to consider and adapt well established SE practices which have been ignored or had a very narrow focus in ML literature [13] [15]. According to [147], software-intensive companies evolve through five stages based on their use of ML. They are (a) Experimentation and prototyping (b) Non-critical deployment (c) Critical deployment (d) Cascading deployment and (e) Autonomous ML components. During the experimentation and prototyping stage, the models do not proceed to real production. In the second stage, ML models are deployed to non-critical functions of the product in companies. As a result, the product remains unaffected even after the failure of ML models. At the third stage, the failure of ML deployment impacts the functioning of the overall product. In the cascading deployment stage as the output of one model is fed as input to the next model, monitoring and ensuring the proper functioning of the system is highly complicated. In the last stage, companies enter into a point where the models are self-capable of monitoring their behaviour and eventually kick-off retraining when required. The immature and early stage of the deployment phase in most of the companies limit their models from proceeding to production.

7.1.3 AI on the Edge: Architecture Alternatives

Most software-intensive companies in the embedded systems domain are trying to move their intelligence from centralized cloud to fully decentralized edges. The main objective of the study was to guide practitioners on what architectural alternatives exist and how these help companies to improve their edge/cloud (re)training activities. For this, we developed a generic framework consisting of five different architecture alternatives

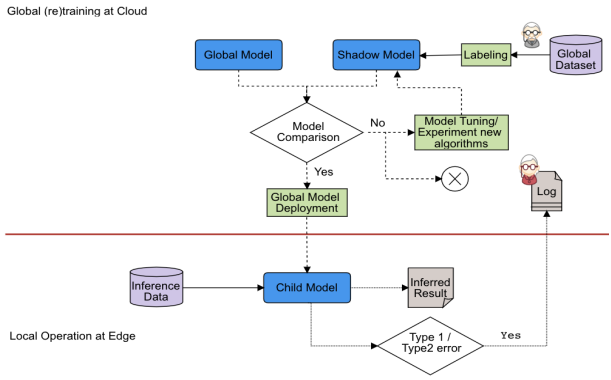

Figure 15: Arch 1: Global Optimum approach [148]

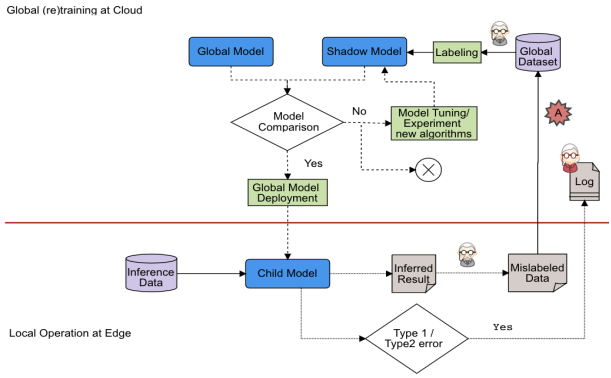

Figure 16: Arch 2: Global Optimum approach + Local Data for actual Global Retraining [148]

ranging from centralized cloud to fully decentralized edge nodes for deploying AI by utilizing the techniques of transfer learning and federated learning [148]. The learnings presented in this study was based on action research in a software-intensive embedded systems packaging company in which we studied different ML/DL use cases. Figure 1-5 illustrates the generic framework consisting of five different architecture alternatives. Below, we describe each architecture.

a) *Architecture 1 - Global Optimum approach*:

In architecture 1, traditional ML/DL systems adopt more of a centralized approach where the global model (re)training takes place in the cloud and the child model is placed in local operation for inference at the edge.

b) *Architecture 2 - Global Optimum approach + Local Data for actual*

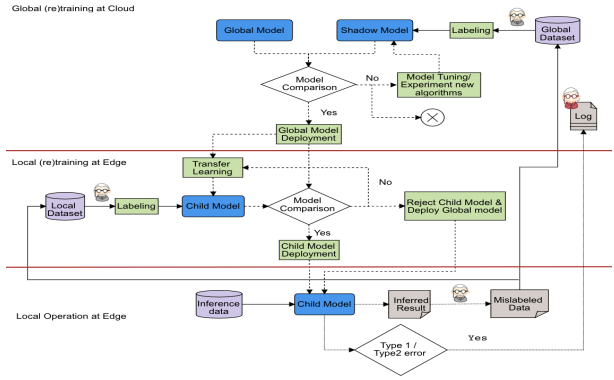

Figure 17: Arch 3: Local Optimum approach + Transfer Learning [148]

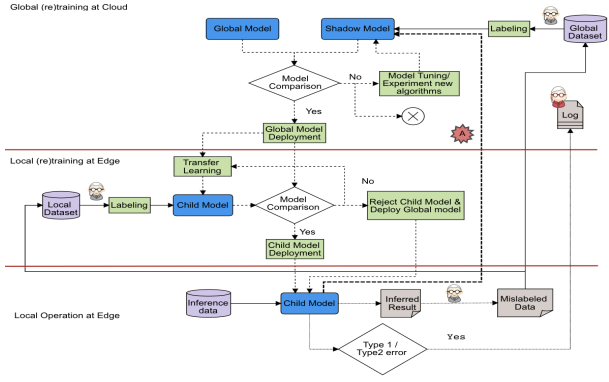

Figure 18: Arch 4: Local Optimum approach + Federated Learning + Local Data for actual Retraining [148]

Global Retraining:

In the second architecture, the global model (re)training takes place in the cloud and local operation at the edge. The primary difference between first and second architecture is that mislabeled data identified as a result of inference is fed back to retrain the global model.

c) *Architecture 3 - Local Optimum approach + Transfer Learning*:

In the third architecture, global model (re)training occurs in the cloud and child model (re)training along with inference at the edge. In addition to all activities stated in the second architecture, the global model utilizes a transfer learning technique to transfer certain layers of the global model to the child model for improving performance.

d) *Architecture 4 - Local Optimum approach + Federated Learning +*

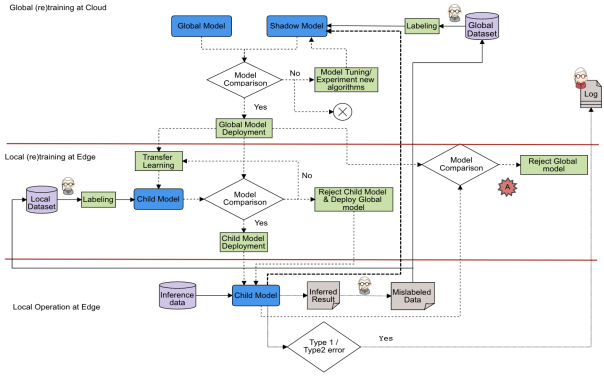

Figure 19: Arch 5: Local Optimum approach + Diverging Deployment [148]

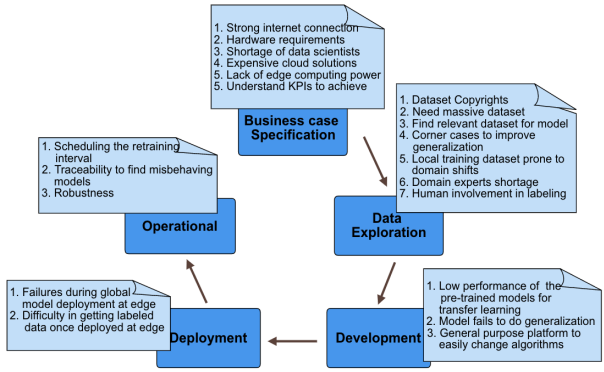

Figure 20: Challenges associated when determining each architecture [**?**]

Local Data for actual Retraining

In fourth architecture, the global model (re)training takes place in the cloud and (re)training of child model at the edge. In addition to third architecture, the learning's from the child model are sent back to the global model for better optimization by utilizing federated learning technique.

e) *Architecture 5 - Local Optimum approach + Diverging Deployment*

In the fifth architecture, the global model and child model (re)training happens in the cloud and at the edge identical to the fourth architecture. In addition, there may be situations in which the global model does not achieve better performance even after retraining with mislabeled data and applying transfer learning and federated learning techniques. In such situations where the global model under performs compared to the child model, companies reject the global model and focus on the child model.

To validate the generic framework, we conducted a qualitative semi-structured interview study with selected practitioners from four additional case companies that were not part of our previous research during that study. From the interviews, we learnt that most of the companies are still deploying their ML/DL models in first and second architecture. We also notice that the majority of companies want to shift to a decentralized approach in the future, even though they lack needed computational capacity currently at the edge. As part of our validation study, we received valuable input describing two more variants of the architectures to the existing architectures that we identified as part of the framework. We also presented various challenges faced by practitioners when selecting a particular architecture for AI deployment in Figure 6.

7.2 Research Method

The research reported in this study is part of a broader research initiative, which has successfully collaborated with fifteen embedded system domain companies. In these companies, we have been conducting longitudinal multi-case study research [152] [112] in a vast range of ML/DL research projects. In this paper, and as a continuation of our previous study of architectural alternatives for AI deployment [148], we expand this study with a focus on factors that help practitioners to select an appropriate architecture for deployment. We adopted a qualitative research [152] in accordance with our research interests in seven software-intensive embedded system domain companies. Based on these findings, we develop an architecture selection framework in which we illustrate how prioritization and trade-offs between these factors result in a specific architecture selection.

7.2.1 Case Company

The research was carried out in close collaboration with seven software-intensive embedded system companies. Although all of the case companies involved in the study represent different industry domains, they are all aiming to increase the decentralization of intelligence as it comes with several benefits and increases value delivery.

To strengthen and advance our learnings from the previous study, we continued our collaboration with new practitioners in five of the previously involved case companies. We also engaged two new case companies

Table 9: The seven case companies involved in our study

Case company	Description
A	A company offering mobility solutions for vehicle development
B	A company providing services, software & infrastructure in communication technology
C	A company manufacturing pumbs & electronics for pumb control
D	A company providing services in healthcare & energy
E	A company offering packaging & processing solution for food products
F*	A company manufacturing and marketing vehicles
G*	A company supplying systems for power generation & transmission & medical diagnosis

to bring in new empirical data and perspectives from additional case companies. We introduced the generic framework of architectural alternatives for AI deployment to all practitioners involved in the study and gathered their insights and reflections. Furthermore, and as a valuable contribution to this paper, we discuss factors from the perspective of practitioners that may or may not influence the selection of an optimal architecture. We provide a short description of the case companies in Table 1. The companies that we did not engage before this study and which are new case companies are marked with * in Table 1. We provide an overview of practitioners involved in the study and the time frame during which the study is conducted in Table 2. In this table, P* denotes interview participants and W* denotes workshop participants.

7.2.2 Data Collection and Analysis

Following the guidance of Runeson and Höst [83], we conducted a multiple case study research by scheduling semi-structured qualitative interview study using an interview guide. In addition to interviews with six case companies, we conducted a workshop with thirty practitioners, including principal key expert for AI for software engineering, software architect,

Table 10: Description of practitioners involved in interview study

Case company	Practitioners		Date
	ID	***Roles***	
A	P1	Vice President	05/06/2020
	P2	Tech lead and Senior Technical Expert	02/07/2020
	P3	Solution Architect	18/06/2020
	P4	Solution Architect	18/06/2020
	P5	ML/AI Engineer	30/06/2020
B	P6	Senior Researcher	23/06/2020
	P7	Data Scientist	23/06/2020
C	P8	Senior Data Scientist	24/06/2020
D	P9	Senior Software Architect	10/06/2020
E	P10	Data Scientist	30/06/2020
F*	P11	Senior Engineer ML & Data science	25/06/2020
G*	W1 - W30	AI Experts	01/07/2020

technical expert, research group head, senior key expert engineer, etc. at one of the case companies. It should be noted that at the beginning of each interview and workshop, we introduced all practitioners to our architectural alternatives for AI deployment. As a follow-up to this, we enquired the practitioners regarding their reflections on factors they consider as relevant in the selection of each architectural alternative. All interviews and workshop were held in English and lasted for one hour. This was conducted during June - July 2020 and notes were taken during interviews and workshop to capture empirical data. They were shared for analysis among the researchers and the investigator triangulation [98] was used to draw different views on the topic under study.

7.3 Case Findings

As part of our previous research based on the action research study and validation study in software-intensive embedded system companies, we empirically derived a set of factors that can potentially influence the selection of architectures for the deployment of ML/DL models. In relation to the follow-up study in this paper, we presented these factors to the practitioners during interviews and workshop, and examined their reflections. As an output of the study, we identify three key factors that help practitioners decide which architecture to select for the deployment of ML/DL

models.

7.3.1 Key Factors

The fundamental principle of key factor analysis is to identify factors that contribute strongly to the variation of different architectures for ML/DL model deployment. Three key factors that play a significant role in selecting an optimal architecture for deployment are (i) Device Cost (ii) Model Performance (iii) Data Privacy. Below, we detail each key factor.
1) **Device Cost:** Based on the interviews and workshop meeting, we find that device cost is perceived to be the most critical factor by a majority of the practitioners. According to them, the cost has the power to guide all other decisions related to the selection of a particular architecture for ML/DL deployment.

> P1: *"Cost is and will always be our number one priority for at least predictable future"*

> P4: *"Cost is always a key aspect in the automotive industry"*

The device cost determines whether the computational capacity can be placed either in the cloud or at the edge. Companies are adopting centralized architecture, mainly because of their cost advantages. A cloud solution enables companies to reduce their hardware, software and infrastructure costs. On the other hand, edge instances require critical management of computation, storage, communication and energy resources. *P1* confirms that the resources available for (re)training in the cloud are completely different from the resource requirements at the edge. One of the workshop participants also points out that if an attempt has been made to deploy AI at the edge, the key question is regarding the cost. Reducing the costs of the edge device leads to a limitation of the number of instances that can be deployed at the edge. A typical instance is that some edge devices are either connected to a battery or a power outlet. As a consequence, a constrained device on the battery will result in less computational capacity and a lack of power. We also note that, as per *P10*, fewer data scientists are required for a centralized cloud approach compared to the edge solution.

> P10: *"It is costly, if the cloud is involved. But finally, you are going to have more costs on the edge, because you need infrastructure and people"*

2) **Model Performance:** Achieving model performance optimization either in the cloud or at the edge can influence the choice of architecture. Companies adopt a centralized approach if it is necessary to achieve some kind of global optimum by combining the entire fleet of devices. In such scenarios, there can be a possibility that individual edge deployments may perform quite badly even though the overall average performance is good. Companies may require global model optimization to ensure that the same global model is deployed everywhere due to certification, liability or other reasons. In such cases, there exists a provision for only inference at the edge. On the other hand, if companies want to achieve local model optimization, they should adopt a decentralized approach. As a result, they can ensure that each deployment at edge location performs as high as possible. It should be noted that if the context of each deployment is unique then the data is also likely to be unique.

> P2: *"If you have a history of more data, then we can go to a decentralized approach"*

Techniques such as transfer learning, federated learning and global model rejection can be used to achieve better performance of local models at each site. For instance, *case company E* has local models to detect defects such as dents, wrinkles, etc., specific to each customer using transfer learning. They maintain a global model for better local optimization.

> P1: *" A very similar situation could arise for people working in manufacturing operations, where different factories have different needs and can do local training."*

3) **Data Privacy:** As expected, all case companies consider privacy to be a challenging factor.

> P7: *"This is an area that we think is quite challenging. Privacy and security is important to us in customer-aspect"*

> P4: *"Not only cost, privacy is another bottleneck"*

The curious fact that we observe from the interviews and workshop is that customers in some countries are not at all concerned about the privacy of their data. They are used to the fact that companies have access to their

private data. For instance, all data related to the plugin hybrid of a country-specific vehicle market must be sent to the government cloud if legally necessary. Although this data is not strictly personal, it can still be used to identify people. In contrast to this, EU residents have even more control over their data than any other countries due to the General Data Protection Regulation (GDPR) policy. It is evident from *P7* that *case company B* often finds privacy to be a limiting factor that stops them from gaining valuable insights from their customers.

> P7: *"We have use cases where we want to maintain central model management but also wants to offer things to our customers so that they can add features/new models"*

According to practitioners *P3* and *P4*, end-users and customers have different perceptions of privacy when it comes to data usage and ownership rights. End-users are most concerned with a good integrity solution in place to ensure that all their data is properly managed. On the other hand, end customers have models hosted for data monetization as they have clear business aspects of how to effectively use the end-user data for making profits.

7.3.2 Architecture selection framework

All selection decisions usually involve trade-offs between factors. If both factors are equally desirable, then we need to balance the benefits of one factor against the other. It is often difficult to set out a clear priority when both factors are important. So all prioritization must be done based on sound analysis. Based on our interviews and workshop meeting, we present a framework in which we outline how prioritization and trade-offs between these factors result in the selection of a certain architecture. The architecture selection framework is illustrated in Figure 7. Different types of lines in the figure represents five different architectural alternatives ranging from centralized cloud to decentralized edges. Prioritization and trade-off between key factors resulting in the selection of a particular architecture is shown in the figure. Below, we present the framework.

Architecture 1 - Global Optimum approach:

In this architecture, if primary importance is given to cost, then there is a need to place (re)training of ML/DL models on the cloud and only provision for inference at the edge.

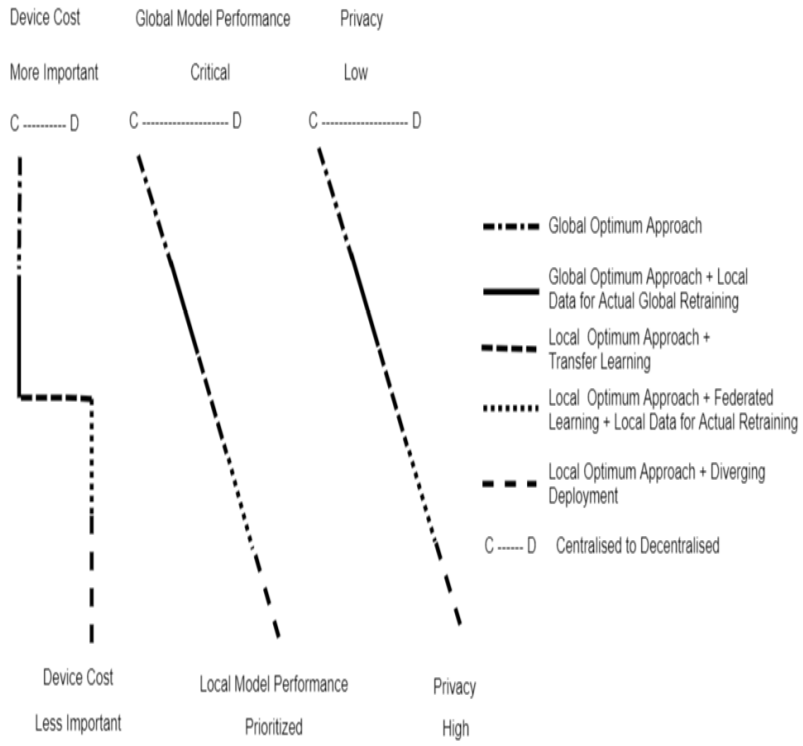

Figure 21: Architecture selection framework

> P1: *"There can be trade-off between what can be done in car and what can be done in cloud in terms of device cost. In most of the cases, this trade-off leads to cloud solution probably"*

If cost is prioritized, optimization of global model performance can be accomplished, but customization requirements at the edge must be loosened to a greater extent.

> P10: *" When thinking about the possibility of continual learning, which is basically the idea to fine tune models at edge, I should go with centralized one if I do not have either the possibility like technology/algorithm unreliable to do that on edge or expertise to fine tune models at edge"*

Given the high computing capacity available in the cloud, customers have to agree on transporting their data to the cloud resulting in cost prioritiza-

tion over privacy. Since there is no provision for data storage or (re)training due to cost constraints at the edge, customers cannot press hardly for data not to be sent to the cloud. Companies can easily prioritize performance and privacy due to the desire of customers/end-users for customization and usability. This tempts them to share and send their data to the cloud solution. This provides great benefits to companies as they can use these data for business profits. In certain situations, privacy concerns over data often hinder the availability of local data for achieving global model performance in companies.
Requirements: Companies can adopt this architecture when they set high importance to device cost, focus on optimization of global model performance and less privacy.

Architecture 2 - Global Optimum approach + Local Data for actual Global Retraining:

The architecture 2 is similar to the architecture 1 in the case of prioritization among factors. Cost limitations lead to more of a centralized approach in this architecture.

> P9: *"If we do not have the capacity for small devices on the edge and we don't have a capacity to do training on them, then we need to take a cost to do training at the cloud"*

> P10: *"If child model does not have intelligence at the edge, we cannot optimize at edge"*

The evident difference between both architectures is that better global model performance optimization can be achieved in second architecture by sending mislabeled data to the global data set for retraining. In this case, performance has to be prioritized over privacy to a large extent. Interestingly, customers tend to prevent misuse of the data they share with others, but on the other hand, they are always willing to share necessary information to obtain satisfactory results.
Requirements: Companies can adopt this architecture when there is a need to provide high importance to device cost, obtain more global model optimization compared to architecture 1 and less privacy.

Architecture 3 - Local Optimum approach + Transfer Learning:

In this architecture, when cost is given less importance and more resources/computational capacity are possible at the edge, then both (re)training of models and inference happens at the edge.

> P2: *"It is likely to have more cost on the edge"*

Model (re)training at the edge after applying the transfer learning technique results in unique model deployment. It will contribute to better optimization of local model performance. Thus paying more costs for customized model performance leads to a trade-off. Customers usually often want high performance at low cost, which results in a need for a balance between cost and performance. Since local model optimization is possible in this architecture, the privacy of customer information can be preserved to a greater extent which again leads to a need for balance between cost and privacy. Data can be stored and processed at the edge as there are no-cost limitations. So there is no need to send customer data to the cloud. In this way, local model performance optimization and high privacy can be easily achieved.

> W2: *"Model segmentation is important"*

Requirements: Companies can follow this architectural alternative when there is less importance to device cost, more focus on optimization of local model performance by utilizing transfer learning and high privacy.

Architecture 4 - Local Optimum approach + Federated Learning + Local Data for actual Retraining:

Architecture 4 is similar to architecture 3 when trying to achieve a feasible balance between different key factors. The main difference is that further local model optimization can be accomplished in this architecture by adopting federated learning in addition to transfer learning. As a result, better model performance can be achieved in the context of paying more cost and preserving high privacy.

> P9: *"We have use cases where we want to have a central management of models, but we also want to offer things to our customers, so that they can add features/add new models"*

Requirements: Companies select this architecture when there is less importance to device cost, need for high local model performance by utilizing

federated learning and transfer learning, and high privacy.

Architecture 5 - Local Optimum approach + Diverging Deployment:

In architecture 5, device cost has no limitations as it demands high local model optimization by rejecting the global model. The risk is to ensure optimization of local model performance even after rejecting the global model. (Re)training of stand-alone edge models results in diverging model deployment in the absence of a global model. A lot of resources are required at the edge to achieve local model optimization which emphasizes on the need to balance cost and performance.

> P10: *"At the end of the day, I would like to end up with decentralized approach. This give a possibility to directly fine tune at edge"*

Selecting this architecture can ensure high privacy because there is no data transfer to the cloud. Although this architecture offers high model performance and high privacy at a high cost, the trade-off is that customers want high performance and high privacy at a low cost. There is a need, therefore, to find a balance between all key factors.

> P10: *"Client does not want to have their data public"*

Requirements: Companies adopt this architecture when the device cost is very less relevant, need high local model performance optimization by rejecting the global model and a high demand for privacy.

7.4 Discussion

In this section, we discuss various trade-offs that need to be made between key factors, as well as additional factors that may or may not influence the selection of a specific architecture.

7.4.1 Trade-offs

All decisions often involve trade-offs. Trade-offs between factors are universal and unavoidable. Prioritizing trade-off is one of the most challenging tasks when selecting a specific architecture for deploying AI on the edge. Mostly, solution architects make these trade-offs in companies and they

do them with a lot of gut instinct. In addition to *P1*, *P11* also agrees with this statement by quoting that

> P11: *"Most of it is done with gut feeling or expertise because they have just seen it for so many times"*

One of the interesting facts is that it is easier to spot people with algorithm knowledge. On the other hand, it is hard to find solution architects who are very good at making these assessments and trade-offs with an overall idea on the best possible solution.

> P1: *"It is not easy to teach these at universities since they tend to focus more on algorithm portion, then they have to work with algorithm for a number of year before you slowly build up the expertise to be able to build up the work. We need solution architects that are data scientists"*

Below, we describe trade-offs that occur during the selection of the generic framework consisting of five different architecture alternative for AI system deployment:

Cost and Performance: During architecture selection, companies often face cost and performance trade-off. In such cases, they need to prioritize between cost and performance benefits. Even though the universal fact is that high cost leads to high performance, people always want to achieve the best performance at a low cost. If the cost is given more importance, then choosing a centralized architecture is the best possible solution.

Cost and Privacy: Companies experience cost and privacy trade-off. Privacy protection incurs higher costs. Even if people are not willing to spend money on high privacy, they wish to safeguard their data. As a result, there exists a need for a balance between cost and privacy.

Performance and Privacy: Practitioners typically prioritize performance and privacy as people are ready to sacrifice their data for better usability and customization. If customers are interested in protecting the privacy of their data, they can choose a fully decentralized approach at the expense of high costs. So, a trade-off between performance and privacy is evident when selecting the optimal architecture.

7.4.2 Additional factors

In addition to the key factors, practitioners provide valuable insights on additional factors that may or may not affect the architecture selection. According to practitioners, the selection of a particular architectural pattern is based on the use case. According to *P3*, if we keep on adding factors that are relevant to architecture selection similar to feature engineering, it will end upon a sub-optimal solution.

> P3: *"An architectural pattern will be chosen based on what is most specific or appropriate for a particular feature"*

> P2: *"It is not super clear as one pattern fits everything, you need to have different kinds of patterns for different features and use cases"*

We note that the selection of a certain architecture for AI system deployment relies heavily on the business of a certain company. It depends on both the device architecture and the kind of computation available in the device architecture within the company. There is a constraint on what you can do in terms of edge computing if the company is not advanced in terms of recent trends. For instance, if a software-defined vehicle has more bandwidth, hardware with computing capabilities and more dynamic and flexible software, then there is a luxury to choose from a lot of architectures.

> P4: *"... so, it depends on where you are in the journey"*

Even if there exists a lot of architectural patterns for deploying AI on the edge, some companies choose centralized approach if the use-case is simple and has less privacy risk. *P11* confirm that priority is given to simple approach that gives the best accuracy when selecting a certain architecture. The cloud solution seems to be the right choice if there is a provision to upload a huge quantity of data. On the other hand, even though 5G is available, it is not feasible to send vast quantities of data to the cloud. In addition to the restriction in the research capability of a company, a lack of expertise to do local optimization at the edge limits customized edge deployment and encourages the same global model deployment at each edge.

> P10: *"If I am going to do 100% sure product, I am going with first architecture, lots of people and company are going for that which makes me think that it works better"*

We notice a disagreement among practitioners as some of them consider moving to edge incurs high cost whereas others consider that it is less expensive than a cloud solution.

> P9: *"Of course cost is important and what we see is that if we can move things to the edge, actually we can reduce cost"*

Based on the insights from interviews and workshop, we observe that some practitioners often trust the support for global model optimization in decentralized architectures, while some practitioners are interested in attempting decentralized architecture by rejecting the global model.

> P2: *"If we discard the model, we are overfitting. We do not want to discard global model, we want an impact of global model"*

As the availability of local data appropriate for model optimization at the edge is limited due to safety or privacy issues, there is a strong chance to end up in a local model representation instead of global model at the cloud. We observe that practitioners foresee a risk of training local models for a long time if the available local data set is a low representative set.

> P5: *"It may depend on how you do training, you could imagine that local dataset for instance is not fully representative or there are faulty datasets and what would you evaluate on them. Is it even feasible to do transfer learning and get a worst outcome on a dataset?"*

> P9: *"Local model has narrow distribution and if you are training it for long time and optimize it and send those parameters back to global model, you have to be pretty careful of what you do, so that you would not get a representation of local model at cloud"*

According to *P2* from *case company A*, logs can be used to improve the explainability of a system. For instance, companies are interested in

knowing why a particular product has been rejected instead of getting “0“s and “1“s.

> P18: *“Explainability is a big plus“*

Explainability is a built-in feature for simple products that can help to minimize human-in-the-loop. Practitioners of *case company A* believe that any type of cloud/edge solution that requires customer interaction tends to be less attractive to them. For instance, avoidance of customer annoyance is crucial in non safety-critical applications. In the other hand, *case company D* finds it very important as customers can add features/new model themselves, which increases value delivery. *Case company B* and *case company E* also support customer interaction.

> P7: *“In Telecom, it is a lot of human intervention. I think it is a necessity since many customers want to have control of the decisions that are taken“*

Scalability can be easily achieved when shifting intelligence to the edge. In contrast to this concept, P1 states that decentralization decreases scalability and maintainability.

> P1: *“Increasing degree of decentralization is likely to make things worse for scalability and maintainability“*

7.5 Threats to validity

Possible threats [62] to validity were considered and minimized in this study whenever applicable. We considered a) Construct validity: Authors and practitioners involved in the study are most familiar with edge/cloud (re)training and inference. Multiple techniques (semi-structured interviews, meetings and workshop) and multiple sources (senior engineers, AI experts, data scientist, etc.,) were used for data collection and validation. This has allowed us to study it from multiple perspectives. b) Internal validity: Threats that can be caused by faulty conclusions that may occur due to bias on the part of authors during the selection or interpretation of the data. To mitigate this, the authors consulted each other in the event of any uncertainty. c) External validity: It refers to the degree to which the generalization of the findings/conclusions can be justified. We conclude that

by extending our previous research to more software-intensive embedded systems companies and use cases helped to mitigate this threat.

7.6 Conclusions

The transition from prototype to the production-quality deployment of ML/DL models is limited in companies. To advance our previous research, we conducted a follow-up study involving interviews and workshops in seven case companies in the embedded systems domain. Based on these findings, we identify three key factors and develop a framework in which we outline how prioritization and trade-offs between these factors result in the selection of a certain architecture. We believe that this framework will accelerate the transition rate of models from prototype to production-quality deployment stage in the companies. In our future research, we plan to expand this study by involving additional case companies and use cases and to further validate our results with practitioners in the field.

8 Architecting AI Deployment: A Systematic Review of State-of-the-art and State-of-practice Literature

This chapter has earlier been published as
M. M. John, H. H. Olsson, and J. Bosch, "Architecting AI Deployment: A Systematic Review of State-of-the-art and State-of-practice Literature" In International Conference on Software Business, pp. 14–29, Springer,2020.

Most embedded system companies have been progressively integrating ML/DL (Machine Learning/Deep Learning) techniques [18] [146] into their systems due to advancements in hardware and big data explosions. Developing, deploying and maintaining a complex ML-based business system is a daunting challenge [15] [126]. Although a significant number of studies on how to design and train models are published each year, there is noticeably less research on how to manage and deploy these models once they have been trained [153]. In addition, the effort needed to go beyond the prototype stage and to deploy and keep it in production is known to be exceptionally high [15]. The need to consider and adapt well-established Software Engineering (SE) practices that have typically been overlooked or had a minor impact on ML literature is important to the real implementation [13] [15] [154]. Hence it is important to advance understanding of how AI models can be deployed, monitored and evolved.

The contribution of paper is threefold. First, we conduct a systematic literature review in which we review the contemporary scientific literature and offer a detailed overview of the state-of-the-art of AI deployment. Second, we review the grey literature and present the state-of-practice and experience of practitioners. Third, we present a framework derived from

current literature for end-to-end deployment process and try to compare and contrast SLR and GLR results. The rest of the paper is set out in six sections. Section II introduces the research methodology employed in carrying out the study. Section III describes the threats to validity. In section IV, we focus on the results of the study. We present the derived framework in section V and discussions in section VI. Section VII summarizes the related works. Finally, section VIII provides conclusions and future work.

8.1 Research Method

The study aims to advance understanding of how to integrate, deploy, monitor and evolve ML/DL models in the context of edge/cloud/hybrid architectures. We believe that this research initiative has the potential to accelerate the transition of ML/DL models from the prototyping stage to the deployment stage within companies. In order to accomplish this objective, the following research questions have been formulated:
RQ1: What is the state-of-the-art regarding the deployment of AI models as published in contemporary scientific literature?
RQ2: What is the state-of-practice in deploying AI models as experienced and reported in grey literature?

In the context of edge/cloud/hybrid architectures, we conducted SLR (RQ1) [155] [156] and GLR (RQ2) [157] [158] to address different and complementary RQs. We restricted our study during the recent time frame from January 1, 2010 to August 31, 2020 in order to obtain recent results, since most companies are currently at the prototyping stage and are slowly moving towards the deployment stage.

8.1.1 Systematic Literature Review (SLR)

The SLR is designed to identify, analyze, and interpret all relevant studies on the topic of interest [155] [156]. This is the proposed methodology for aggregating empirical studies. To answer RQ1, we have defined search strings that included both research objective as well as research question terms, as suggested by [155]. We queried five popular scientific libraries to retrieve the relevant studies. The search queries and the retrieved studies of each scientific library are listed in Table 1. The retrieved studies were integrated and exported into the excel sheet. As an inclusion criteria, we defined (a) Studies that report the deployment of ML/DL models within

Table 11: Search query used to retrieve relevant studies from selected libraries

Scientific Library	Search Query	Filters
IEEE Xplore	“All Metadata": software AND (deploy* OR production) AND (integrat* OR inference OR serv* OR monitor* OR scale OR evol*) AND (edge OR cloud OR hybrid) AND (“machine learning" OR “deep learning")	Conferences & Journals
ACM Digital Library	software AND (deploy* OR production) AND (integrat* OR inference OR serv* OR monitor* OR scale OR evol*) AND (edge OR cloud OR hybrid) AND (“machine learning" OR “deep learning")	PDF
Scopus	TITLE-ABS-KEY (software AND (deploy* OR production) AND (integrat* OR inference OR serv* OR monitor* OR scale OR evol*) AND (edge OR cloud OR hybrid) AND ("machine learning" OR "deep learning")	Conferences & Journals
ScienceDirect	software AND (deployment OR production) AND (edge OR cloud OR hybrid) AND (“machine learning" OR “deep learning")	Conference & Journals
Web of Science	TS=(software AND (deploy* OR production) AND (integrat* OR inference OR serv* OR monitor* OR scale OR evol*) AND (edge OR cloud OR hybrid) AND (“machine learning" OR “deep learning"))	Article or Proceeding paper

the context of edge/cloud/hybrid architectures. We excluded (a) Duplicate versions (b) Not written in English (c) Not a peer-reviewed scientific research and (d) Not available electronically through web. Figure 1 outlines the overall SLR process adopted in the study. [159] - [160] represent primary studies.

8.1.2 Grey Literature Review (GLR)

Researchers are increasingly using GLR in their study. As reported in [157], although the SLR and the Systematic Mapping Study (SMS) provide comprehensive descriptions of the state-of-the-art, they typically lack the state-of-practice. This state-of-practice is critical in a more practice-oriented field of study such as SE. There is also a disconnect between studies reported by researchers and practitioners, as practitioners hardly involve in and display interest in scientific literature publications. Using GLR [158] in studies can (a) Eliminate publication bias (b) Offer contextual knowledge

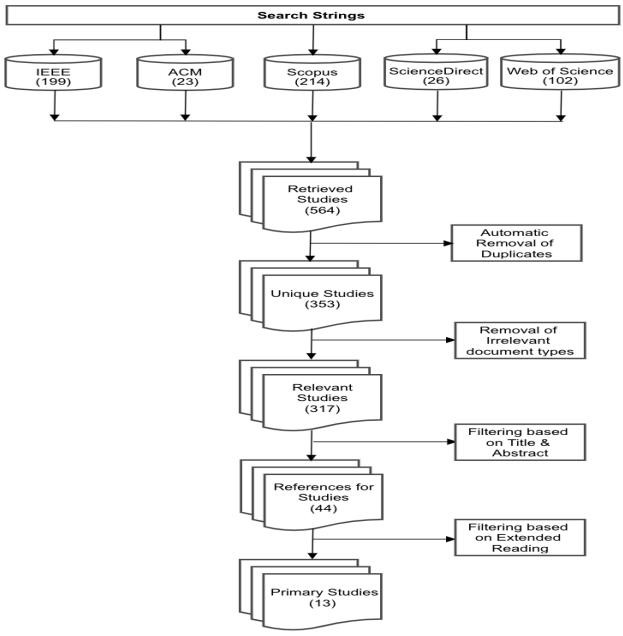

Figure 22: Research Process for SLR

(c) Allow SE researchers to understand solutions to a specific existing SE challenge, perform evaluations and identify areas where further evaluation and improvement are required.

Grey literature may offer practitioner perspectives on key topics relevant to practice and research, as well as encourage the voice of practitioners. Grey literature is suitable for this study, as it reflects the state-of-practice and experience of practitioners when deploying AI into cloud/edge/hybrid architectures. As part of conducting the GLR, we included studies that (a) Focus on integration, deployment, operationalization and evolution of ML/DL models in companies within limited time frame (b) Written in English (c) PDF file format and (d) Included documents from companies by filtering site as ".com". We excluded (a) Peer-reviewed scientific articles and (b) Other sources of information such as blogs, posts, etc. to increase the reliability of results. We explored Google search engine using the below query, which resulted in sixteen studies referring to RQ2. Among these sixteen studies, we selected six studies [161] - [162] that benefit our research.
(deploy* OR production) AND (integrat* OR inference OR monitor* OR

evol* OR scal* OR operation*) AND (machine learning OR deep learning) AND (edge OR cloud OR hybrid) site:.com filetype:pdf

8.2 Threats to Validity

Potential threats to validity are taken into account and minimized in the study [62]. They are as follows: (a) Construct Validity - Enhanced by conducting SLRs and GLRs to identify the state-of-art and the state-of-practice to better study AI deployment from multiple perspectives. A considerable threat is that two of the thirteen primary studies extracted from SLR included studies that had the participation of practitioners in companies. Hence, there could be a potential possibility of conflict between the results of SLR and GLR. This threat is inevitable, because some practitioners seldom publish their work in peer-reviewed scientific articles. To accurately collect primary studies using SLR and GLR, we defined search process, search query, inclusion and exclusion criteria. (b) Internal Validity - Threats caused by author bias when selecting and interpreting data leads to internal validity threats. The SLR results are complemented by the GLR results to provide a clear overview of the subject under study. (c) External validity - Results can be extended to all software-intensive companies deploying ML/DL models as the study encompasses both state-of-art and state-of-practice reported in the literature.

8.3 Results

Based on the state-of-the-art and state-of-practice reported in SLR and GLR for deploying AI in the context of edge/cloud/hybrid architectures, we extract the practices and challenges. According to the findings from the literature, they are grouped into five phases i.e. Design, Integration, Deployment, Operation and Evolution. Each phase consists of two tasks. These phases have been chosen based on the literature review. In addition, most phases represent the keywords used in our search query. The phases and tasks extracted from SLR and GLR are shown in Table 2 - 4. The hybrid architecture referred to in the study consists of a combination of cloud and edge architectures to deploy ML/DL models.

Table 12: Practices and challenges extracted from SLR and GLR [1/3]

Phases	Tasks	Practices	Challenges
2*Design	Validation	- Execute validation techniques: modelling, training and test error and cross-validation [163] - Terminate training process and release occupied computing sources [164] - Understand collected features with effect on outcome [165] - Compare experiments and run burn-in tests [166] - Plan model deployment [165]	- Data scientists need new ways of knowledge sharing [165] - Data scientists prefer to develop models alone [165]
	Tracking	- Track models, dependencies [165], experiments [166], versions [167](eg: GitHub hash tags [165]), etc. - Maintain registry for model status and artifacts [166] [165]	- Failure to version models leads to undesired situations [166]
2*Integration	Resource Discovery	- Store models in a single format for ease of use [167] - Execute resource discovery and set up resources [168]	- Difference in prototype and production environment in terms of hardware, OS, library, etc. [166] [165]
	Rewrite/ Package	- Two ways for integrating models: (a) Rewrite from data analysis to industrial development language (b) Equip with web interface [169] - Rewrite model for integrating to reports/applications or to share insights and prediction with analytic products [165] - For web interface, package image (eg: docker image [159] [166]) with frameworks and libraries [170] [169] [171] - For reproducibility, use standard run time and configuration files [167] - Technology for packaging different applications: (a) Containers [159] [167] [168] [170] [169] [172] (b) Serverless computing [170] (c) Hypervisor-based virtualization [168] - Existing containerization solutions: (a) Docker [159] [166] [167] [173] [168] [170] (b) Singularity [167] [173] (c) LXC [170] - Abandon usage of hypervisor-based virtualization [168] - Provide integration with existing data infrastructure and ensure data access and storage [166] - Apply compression before deploy, if needed [173] [171] - Select ML solution fully integrated with databases to reduce efforts [165]	- Difficulty in integrating ML models into existing or new applications [165] [162] - Lack of docker containers optimized for accelerator [167] - Implementing same model in different frameworks need time and efforts [167] [174] - Support code updates due to change in API framework [167] - Converted model performs bad compared with model implemented using native API framework [167]

Table 13: Practices and challenges extracted from SLR and GLR [2/3]

Phases	Tasks	Practices	Challenges
2*Deployment	Target Environment	- Host model to invoke and get predictions [174] - Run workloads in [165] [172] [170] (a) Public cloud (b) On-premise (c) Hybrid cloud (d) Edge - Use cloud for compute-intense tasks, otherwise opt edge [175] [170] - If edge is incapable to process, forward the request to cloud [170] - Minimize deploy cost while giving good QoE to users [170]	- Difficulty in deploying neural networks in edge devices due to limiting computation and memory resources [176] - Most DL packages (Caffe, TensorFlow, etc.) focus on cloud and not on edge [176] - Human experts lack GPU configuration knowledge for device placement [160]
	Launching	- Decompress unpack image [173] - Expose deployed application as services to external users via communication channel [159] [168] - Provide REST API to pipeline [169] [166] [176] [165] (e.g.flask API [170]) - Need support from data scientists, engineering teams [166], ML engineers [171] data engineers, domain experts, infrastructure professionals and end users [165] [174]	- Complicated process to deploy and configure DL framework and AI models execution on edge [176] - EU financial institutions are not yet allowed to move to cloud by EU regulators - Lack of skills among data scientists to deploy REST API endpoint [165] - Inability of ML models to stay on laptops of data scientists [165]
2*Operation	Inference	- Ensure proper deployment of inference pipelines to pass, pre-process, predict and post-process raw data to serve batch and real-time requests [165] [174] - Provide simple API for serving prediction [174] - For processing online inference request for features computed offline [166], take values: (a) End -of-day (b) Start-of-day - Needs to be elastic in response to traffic changes [166] [163] [174] - Ensure good data is used in both prototyping and production setup [165] - For latency critical and accuracy insensitive tasks, use low-latency mode with fog configuration else use high accuracy mode [175] - Response time, jitter and power usage [175] is: (a) Low, if send to fog nodes (b) High, if send to cloud	- Companies find difficulty in operationalizing [165] - Ensure data consistency across environments of offline training and online inference [166] [165] - Scalability and security are major concerns [166] [165] - Network latency is the bottleneck in end-to-end latency [170] - Difficulty in serving each user with separate edge computing instance when number of users increase - To reduce latency, lot of packages sacrifice memory for execution on edge [176] - Inference cost goes up when multiple models are deployed with different endpoints [174] - Mostly GPU instances are oversized for inference [174]

Table 14: Practices and challenges extracted from SLR and GLR [3/3]

Phases	Tasks	Practices	Challenges
	Monitoring	- Ensure monitoring system for querying, visualising and deeper understanding of metrics and event logging [159] - Monitor status and performance [166] and compare to business expectation [174] -Deploy initial model and boost traffic incrementally if it works well [165] - Determine concept drift, [165] memory consume by production environment and errors returned by models - Measure inference accuracy of deployed model to ensure data drifts noticed and actions taken [174] -Automatic roll back and forward capability to recover from degraded model performance [174] [172]	- ML is deemed valuable by companies until it begin to improve profit [165] - Models performing well during training might not perform well during production [165]
2*Evolution	Retrain	- Employ Agile, DevOps-style work flows [163] - Ability of teams to re-evaluate the models quickly and use CI/CD to keep updated and prevent decay [163] [174] - Retrain model when changes in model performance occur [174] [165] - Apply automation to trigger model retraining which reduces effort and human error [174] - Retraining model on personal data outperform general models - Update model with latest data instead of retraining on historical data [164] - Create more model instances by: (a) Periodically [164] (b) Accuracy lower than threshold	- Most early users of ML models (a) Do not refresh [165] or replace models when accuracy degrades (b) Do so at each fixed intervals rather than continuously
	Redesign	- Choose tools and a broad set of diagnostics and monitoring ability to check if model under-performs and redesign from scratch [165] - Refine and scale solution as new technology is available [163] [165] - Use blue/green deployment, canary deployment technique to deploy new version of model to production [174] - Deploy different versions of same application based on different needs. For instance, by utilizing A/B test [159] [174] or take a multi-armed bandit approach [165]	

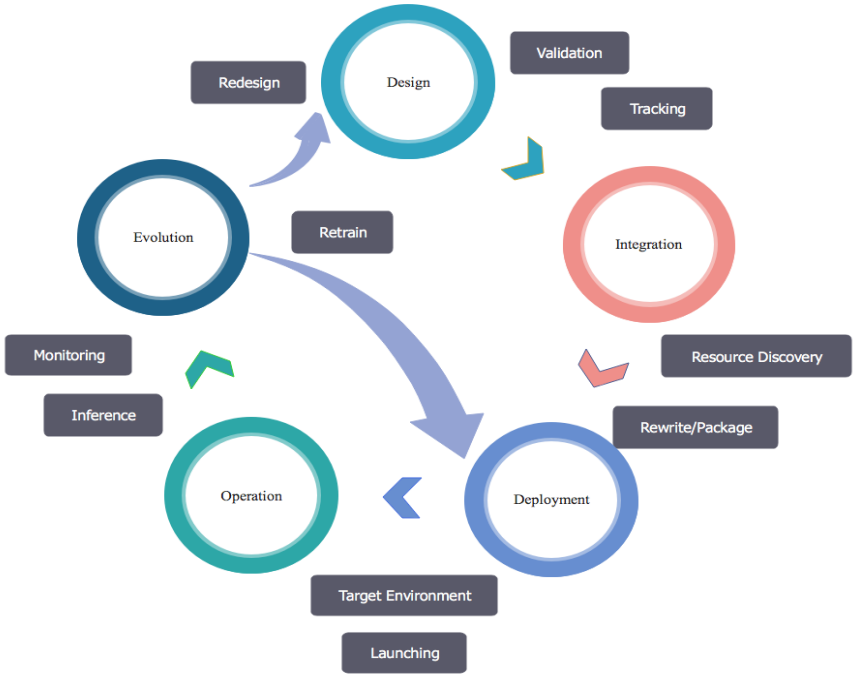

Figure 23: End-to-end Deployment Framework

8.4 Framework

Based on the insights gained by synthesizing practices extracted from SLR and GLR, we derive a framework to facilitate the end-to-end deployment process of ML/DL models. Figure 2 shows an illustration of the framework. The overall framework is structured into five phases and consists of two tasks for each phase. They are: (a) Design - Validation & Tracking (b) Integration - Resource Discovery & Rewrite/Package (c) Deployment - Target Environment & Launching (d) Operation - Inference & Monitoring (e) Evolution - Retrain & Redeploy. Below, we detail each of the phases:
A. **Design**: When the models are properly *validated* [163] and offer confidence in results, the model is ready for placing into production. At this stage, the training process is terminated and all associated computing resources are released [164]. Before bringing the models into production, ensure that necessary *burn-in tests* are carried out to avoid initial model failures when put into production [166]. *Proper planning* of deployment process [165] can ensure a smooth transition from prototyping to deployment phase. Models, dependencies, artifacts, etc. need to be tracked and

versioned to ensure reproducibility [165] [166]. For instance, GitHub hashtags can be used for code versioning [165]. It is highly recommended to maintain a registry containing entries for status of model life cycle and artifacts [166] [165].

B. **Integration**: After validating and versioning the models properly, save them for reuse. Next, *discover and set-up* the necessary resources to put the model into production. Models can be incorporated into the application logic in two primary ways. These are: (a) *Rewrite models* from the data analysis language (for instance, R or Python) to industrial development language (for instance, Java or C++). Models are often rewritten to integrate into applications or reports or to share knowledge and predictions with analytical products. (b) Provide *web-interface* to the models. In the latter case, the model images are *packaged* with the required frameworks and libraries. One of the most popular technologies for packaging is *containerization*. It is important to ensure that the model integrate well with existing infrastructure and also verify appropriate data access and storage. Models are *compressed* based on use case requirements and resource availability prior to deployment.

C. **Deployment**: The compressed models are *decompressed* and unpacked in the target environment [173], if necessary. These target environments can be cloud, edge or hybrid cloud-edge collaboration, according to the use case [165] [172] [170]. For instance, more computation-intensive and less safety critical tasks need to be migrated to the cloud for processing. In contrast, process all latency critical requests at the edge device [175] [170]. The deployment phase deals in particular with the *initial model deployment*. Proper consideration should be devoted to minimize the *deployment costs as low* as possible while at the same time guaranteeing *quality of experience* to end-users. The deployed application is provided to users through a communication channel [159] [168]. Introducing the model into production demands *close collaboration* between data scientists, ML engineers, engineering teams, data engineers, domain experts, infrastructure professionals, etc. [166] [171] [165] [174].

D. **Operation**: Companies perceive operationalization as the most challenging phase. Ensure proper deployment of models, inference pipelines, monitoring and event logging mechanisms before serving models [165] [174]. In this phase, the deployed model consumes raw data to serve either *batch or real time inference requests* [165] [174]. Model performance can be improved by deploying the initial model and slowly increasing the

traffic to the model instead of allowing it to serve all requests at the beginning [165]. It should be noted that most of the GPU resources used in training DL models are oversized for inference [174]. The deployed model undergoes *continuous monitoring* to determine data drifts, concept drifts, returned errors from model, etc. [165]. As model performance degrades, introduce *roll back mechanisms* for quicker recovery [174] [172].
E. **Evolution**: The evolution phase deals with subsequent model deployment. As the models are placed in production, performance degrades over time [174] [165]. If so, select one of the two mechanisms: (a) Retrain (b) Redesign. Teams may employ CI/CD (Continuous Integration/Continuous Deployment) to keep the model up-to-date. Provisions to automatically trigger retraining will lessen both human errors and efforts [174]. Instead of retraining the entire model, updating the model with the latest information [164] is a good option. Techniques such as A/B test [159] [174] and multi-armed bandit [165] can be selected for deploying different versions of the same model to determine the best performing model. On the other hand, blue/green deployments can be used to deploy new model versions in production [174]. An iteration is triggered from the evolution phase to the deployment phase when it is necessary to retrain the model. If redesigning the model is appropriate, an iteration is triggered from the evolution to design phase where we experiment with the current models for better performance or from scratch [165]. Teams that evaluate model performance needs to be very quick to prevent model decay as soon as possible [163].

8.5 Discussion

The study highlights the fact that ML/DL model deployment is gaining momentum over the last two to three years, since most primary studies cover the period 2017 - 2020. In addition, only two of the thirteen SLR studies involve participation from companies. This emphasizes that GLR is a valuable source to advance our knowledge about practices and challenges practitioners face in companies. Below, we compare and contrast SLR and GLR findings to provide further insights on AI deployment and illustrated in Figure 3.
Compare Findings: Below, we compare the findings of literature review:
A. *Support among Professionals*: Both SLR and GLR sources suggest that successful deployment of ML/DL models *requires support* from data scientists and other experts such as data engineers, domain experts, infrastructure professionals and end-users. For instance, lack of data scientist

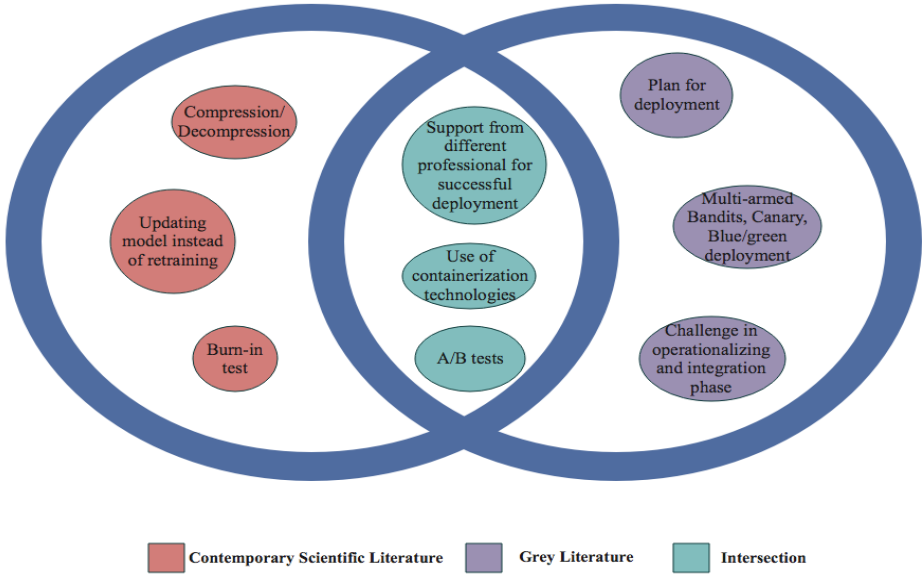

Figure 24: Comparison/contrast between Literature Review Findings

skills to deploy REST API slows down the deployment process. Deployment can also be postponed due to prioritization and limitation of experts.
B. *Use of Containerization Technologies*: Containerization has been identified as the most popular packaging technology. Implementation of same model in different frameworks result in a loss of time and effort. For instance, models often need to be completely rewritten in production preferred languages. Since companies only find ML/DL valuable until they start producing value and profit, it is *not acceptable to delay production* because they want their product to reach markets more rapidly. This is perhaps one of the reasons for increasing demands for containerization.
C. *A/B Test*: Once the model is deployed into production, it is obvious that both SLR and GLR sources recommend adoption of A/B testing technique *to deploy multiple versions* of the best performing model. It can be concluded that fraction of academia projects that proceed into production utilize A/B testing. Besides A/B test, GLR confirms the use of other techniques for deploying new versions. For instance, blue/green deployment.

Contrast Findings: Below, we contrast findings between SLR and GLR:
A.*Compression/Decompression*: Models are *compressed* prior to deploy-

ment and *decompressed* after deployment as stated in SLR based on use case requirements and resource constraints. This is generally used for deploying models at edge by making minimal compromise on accuracy.
B.*Updating Models*: According to SLR, we find technique of *updating* models when the performance degrades with entire historical data *instead of retraining*. For instance, update the model every two hours with the latest up-to-date information or as performance degrades. This technique suits to non safety-critical applications where accuracy is insensitive.
C.*Burn-in Test*: Based on SLR, burn-in tests are executed before real deployment to *stop initial model failures* when deployed on target devices. On the other hand, as per GLR, the models are put in the pilot phase prior to deployment. Although burn-in test is identified from SLR in our study, it is part of one of the two scientific peer-reviewed company-based paper.
D.*Plan for Deployment*: In accordance with GLR, companies always plan beforehand for deployment compared to SLR. Studies show that companies have difficulty in integrating models into existing or new applications, deploying models on edge devices, operationalizing models, etc. Therefore, *proper planning* makes the deployment process even easier with less overhead and resource utilization.
E.*More Deployment Techniques*: Although A/B test is widely used in companies, techniques such as multi-armed bandits, canary and blue/green deployments are employed in companies based on GLR to deploy new model versions. These techniques are absent in contemporary literature, which may be due to the immaturity of deployment process employed in academia.
F.*Challenges in Operationalizing phase*: GLR provides more information about different practices, techniques and challenges in relation to monitoring and evolution compared to SLR. Most models in scientific literature experience only initial deployment and are not constantly replaced/refreshed as performance degrades over time.

8.6 Related Work

ML benefits can only be harnessed when models move from prototyping to deployment stage [177]. According to [2] [9], deployment is an underestimated area where companies are struggling with model testing, trouble shooting, glue code for running the system, and, monitoring and logging mechanisms. Moreover, ML/DL model deployment demands significant change in the overall system architecture as it needs to be integrated to

a software-intensive system. In addition to the components that care for model deployment in production, it is recommended that additional components be provided for model validation prior to deployment, model evaluation and monitoring [178] [179]. Deployed models are dedicated to serve real data [180]. If the model fail to meet expectations, roll back and restore the previous model version. Federated learning requires a specialized deployment framework to address changes in edge [60].

8.7 Conclusions

Most companies are placing lot of models into production compared to previous years. To better understand design, integration, deployment, operation and evolution of AI models, we analyze both SLR and GLR in the context of edge/cloud/hybrid architectures. Based on these findings, we list various practices and challenges practitioners face when deploying ML/DL models. We derive a framework on the basis of literature review for the end-to-end deployment process and also attempt to compare and contrast the findings of SLR and GLR. We look forward to validate the framework in various software-intensive domain companies to better understand the deployment process.

9 Towards MLOps: A Framework and Maturity model

This chapter has earlier been accepted as
M. M. John, H. H. Olsson, and J. Bosch, "Towards MLOps: A Framework and Maturity model," In 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA), IEEE, 2021.

Machine Learning (ML) has a significant impact on the decision-making process in companies. As a result, companies can save significant costs in the long run while ensuring value for their customers [181] and also enabling fundamentally new ways of doing business. To improve value creation and automate the end-to-end life cycle of ML, data scientists and operations teams are trying to apply DevOps concepts to their ML systems [182] in companies. DevOps is a "set of practices and tools focused on software and systems engineering" [41] with close collaboration between developers and operations teams to improve quality of service [183]. ML models embedded in a larger software system [44] are only a small part of the overall software system, so the interaction between the model and the rest of the software and its context is essential [15]. From literature it is apparent that ML processes are often not well integrated with continuous development and production in practice [182].

Despite the popularity of ML, there is little research on MLOps because it is a recent phenomenon. To advance understanding of how companies practice MLOps, including collaboration between data science and operations teams, we use a Systematic Literature Review (SLR), a Grey Literature Review (GLR), and a validation study in three case companies. The paper makes three contributions.

- We conduct a SLR and a GLR literature review to present the state-of-the-art regarding the adoption of MLOps in practice and derive a framework from the reviews
- We present a maturity model with different stages in which companies evolve during MLOps adoption
- We validate the framework and map the three case companies to the stages of the maturity model

The remainder of the paper is organized as follows: Section II describes the background of the study, Section III describes the research methods used and Section IV addresses the threats to validity. Section V summarizes the findings from the literature review. Section VI describes the MLOps framework and maturity model. Section VII describes the validation study conducted in three case companies and Section VIII discusses the results. Section IX concludes our study.

9.1 Background

This section discusses DevOps, DevOps application on the ML systems (referred to as MLOps), and the challenges associated with it.

9.1.1 DevOps

DevOps [41] aims to "reduce the time between committing a change to a system and the change being placed into normal production while ensuring high quality" [42]. The goal is to merge development, quality assurance, and operations into a single continuous process. The key principles of DevOps are automation, continuous delivery and rapid feedback. DevOps requires a "delivery cycle that involves planning, development, testing, deployment, release and monitoring as well as active cooperation between different team members" [41].

Continuous software engineering (SE) refers to iterative software development and related aspects like continuous integration, continuous delivery, continuous testing and continuous deployment. Continuous SE enables development, deployment and feedback at a rapid pace [184] [185] and is divided into three phases: a) Business strategy and planning, b) Development and c) Operations. Software development activities such as continuous integration (CI) and continuous delivery (CD) support the oper-

ations phase. In CI [184], team members of software-intensive companies often integrate and merge development code to have a faster and more efficient delivery cycle and increases team productivity [185]. This facilitates the automation of software development and testing [186]. CD ensures that an application is not moved to the production phase until automated testing and quality checks have been successfully completed [187] [188]. It lowers deployment risk, cost and provides rapid feedback to users [189] [190].

9.1.2 MLOps

With the successful adoption of DevOps, companies are looking for continuous practices in the development of ML systems. To unify the development and operation of ML systems, MLOps [44] extends DevOps principles [191]. In addition to traditional unit and integration testing, CI introduces additional testing procedures such as data and model validation. From the perspective of CD, processed datasets and trained models are automatically and continuously delivered by data scientists to ML systems engineers. From the perspective of continuous training (CT), introduction of new data and model performance degradation require a trigger to retrain the model or improve model performance through online methods. In addition, appropriate monitoring facilities ensure proper execution of operations.

9.1.3 Challenges associated with MLOps

In our own previous research [116], we identified a number of challenges In our own previous research [116] [148], we have identified a number of challenges when it comes to the business case, data, modeling and deployment of ML or Deep Learning (DL) models. These include high AI costs and expectations, fewer data scientists, need for large datasets, privacy concerns and noisy data, lack of domain experts, labeling issues, increasing feature complexity, improper feature selection, introduction of bias when experimenting with models, highly complex DL models, need for deep DL knowledge, difficulty in determining final model, model execution environment, more hyperparameter settings, and verification and validation. It also includes less DL deployment, integration issues, internal deployment, need for an understandable model, training-serving skew, end-user communication, model drifts, and maintaining robustness. Some of the challenges in MLOps practice [44] include tracking and compar-

ing experiments, lack of version control, difficulty in deploying models, insufficient purchasing budgets and a challenging regulatory environment.

9.2 Research Method

The main objective of the study is to identify the activities associated with the adoption of MLOps and the stages in which companies evolve as they gain maturity and become more advanced. To achieve this objective, we developed the following research questions:

- RQ1: What is the state-of-the-art regarding the adoption of MLOps in practice and the different stages that companies go through in evolving their MLOps practices?
- RQ2: How do case companies evolve and advance their MLOps practices?

We performed a SLR [155] [156], a GLR [157] [**?**] and a validation case study [192] to address the two RQs.

9.2.1 SLR and GLR

The goal of the SLR is to find, examine and interpret relevant studies on the topic of interest [155] [156]. To answer the RQs, we defined search strings according to [155] and searched five popular scientific libraries. Figure 1 shows an overview of the SLR and the GLR process that was used in this study. We integrated and exported relevant studies into an Excel spreadsheet for deeper analysis. In SLR, we included conference and journal studies that reported MLOps. On the other hand, we excluded studies that were duplicate versions, published in a language other than English, were not peer-reviewed, and were not available electronically on the Internet.

We conducted the GLR [157] to provide a detailed description of the state-of-practice and practitioner experiences in adopting MLOps. Compared to the SLR, the GLR provides the voice of practitioners on the topic under study. In GLR, we included studies in the Google Search that address MLOps, published in English in PDF format and documents from companies by filtering the site under the domain name ".com". To improve the reliability of the retrieved results from the GLR, we excluded

peer-reviewed scientific articles and other sources of knowledge such as blogs, posts, etc.

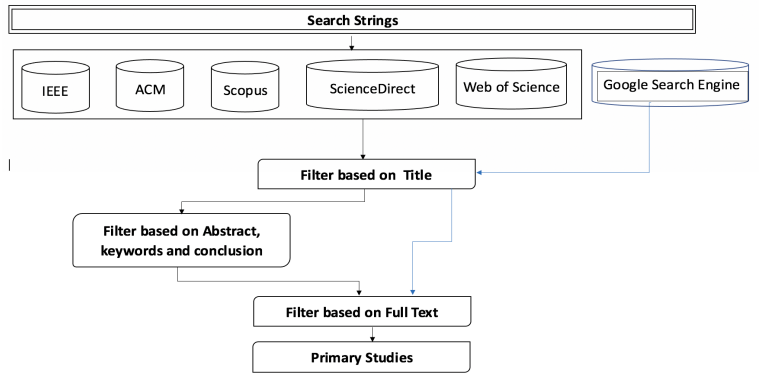

Figure 25: Overall SLR and GLR process used in the study

For the SLR and the GLR, we used the search query as "MLOps" OR "Machine Learning Operations" and restricted the search to the period between January 1, 2015 and March 31, 2021. The time interval was chosen because the term MLOps is prevalent after the concept "Hidden Technical Debt in Machine Learning Systems" [15] in 2015. Based on the SLR and the GLR, we shortlisted 6 SLR ([193] - [194]) and 15 GLR [195] - [196] studies. Based on these studies, we developed an MLOps framework and various stages that companies take in evolving MLOps practices.

9.2.2 Validation Case Study

Following [197], we conducted a validation study to map companies to the stages in the maturity model derived from literature reviews. Case study methodology is an empirical research approach based on an in-depth study of a contemporary phenomenon that is difficult to study separately in its real-world environment [112]. In SE, case studies are used to better understand how and why SE was done and thus improve the SE process and resulting software products [72]. Throughout the validation study, we worked closely with practitioners in each case company. Table 1 provides a brief description of each case company, the practitioners (P*, W*, M*, and S* represent interview, workshop, meeting, and stand-up meeting participants respectively) and their roles.

Case Companies: We present three case companies and use cases that

Table 15: Description of practitioners in the validation study

Case Company	Practitioners	
	ID	***Roles***
Telecommunications	P1, W1, S1	Senior Data Engineer
	P2, W2, S2	Data Scientist
	P3, W3, S3	Data Scientist
	P4, W4, S4	Data Scientist
	W5, S5	Senior Data Scientist
	W6, S6	Data Scientist
	W7, S7	Software Developer
	W8, S8	Software Developer
	S9	Operational Product Owner
	S10	Sales Director
Automotive	W9	Expert Engineer
	W10	Expert Engineer
Packaging	M1	Solution Architect
	M2	Data scientist

were investigated in each company as part of our validation study.
1. Hardware Screening: The telecommunications company predicts faults in hardware to minimize the amount of hardware returned by the customer for repair. In this use case, they focus on a) Returning defect-free hardware back to the customer b) Sending defective hardware to the repair center.
2. Self-driving Vehicles: The company that manufactures vehicles strives to provide autonomous transportation solutions. The main use case is self-driving vehicles to increase the productivity. The company also needs to ensure that the failure rate is low in these safety-critical use case.
3. Defect Detection: The packaging company provides packaging solutions as well as machines to customers. One of the main use cases is the detection of defects in finished/semi-finished packages.

Data Collection and Analysis: For data collection, we used interview studies, workshops, meetings and stand-up meetings in companies. They were held in English via video conferencing. All interviews lasted 45 minutes, workshops and meetings lasted 30 minutes to one hour, and daily stand-up meetings lasted 15 minutes. We validated the MLOps framework

in case companies and present different stages that companies go through when implementing MLOps. Transcripts from interviews and notes from workshops, meetings and stand-ups were used to capture empirical data. Later, they were shared with the other authors by primary author for detailed analysis. We applied elements of open coding to analyse and categorize collected empirical data [97]. In order to obtain different perspectives on the topic under study, triangulation was used [98].

9.3 Threats to validity

Potential validity threats were considered and minimized in this study [62]. Construct validity was improved by considering information from SLR, GLR and the validation case study. Authors and practitioners involved in this study are well versed in MLOps. Multiple techniques (semi-structured interviews, workshops, meetings, and stand-up meetings) and multiple sources (senior data engineer, data scientist, software developer, expert engineer, etc.) were used to collect and validate empirical data. Internal validity threats caused by faulty conclusions due to primary author bias in data selection or interpretation are mitigated by consulting with other two authors. By extending our research to additional case companies, generalization of the results can be justified and thus external validity can be mitigated.

9.4 Literature Review Findings

Based on the SLR and the GLR, we extract insights from the literature to give an overview of the state-of-the-art of MLOps in practice. They are divided into three parts: a) Data for ML Development b) ML Model Development and c) Release of ML Models. Below, we discuss each part in detail.

9.4.1 Data for ML development

Aggregate heterogeneous data from different data sources [198] [199] [200], preprocessing [201] and extracting relevant features are necessary to provide data for ML development. Later, the features are registered in a feature store [202] which can be used for development of any ML models [202] and used for inference when deploying the model. Also, the

data points are stored in the data repository [203] after versioning. The data collected from various sources has to be properly stored and managed. Data anonymization and encryption [201] should be performed to comply with data regulations (e.g. GDPR [123]).

9.4.2 ML Model Development

In ML model development, provisions should be made to run experiments in parallel, optimize the chosen model with hyperparameters, and finally evaluate the model to ensure that it fits the business case. After versioning, the code is stored in the code repository [202] [193]. The model repository [203] keeps track of the models that will be used in production, and the metadata repository contains all the information about the models (e.g., hyperparameter information). Data scientists can collaborate on the same code base, which also allows them to run the code in different environments and against a variety of datasets. This facilitates scaling and the ability to track the execution of multiple experiments and reproducibility [195].

9.4.3 Release of ML models

To release ML models, package [200], validate [200] and deploy models [55] to production [200]. When deploying a model to production, it has to be integrated with other models as well as existing applications [204] [200]. When the model is in production, it serves requests. Despite the fact that training is often a batch process, the inferences can be REST endpoint/custom code, streaming engine, micro-batch, etc. [205]. When performance drops, monitor the model [200] and enable the data feedback loop [200] to retrain the models . In a fully mature MLOps context, perform continuous integration and delivery by enabling the CI/CD pipeline and continuous retraining through CT pipeline [200] [199].

From the literature review, we see that successful AI/ML operationalization ensures a safe, traceable, testable, and repeatable path for developing, training, deploying, and updating ML models in different environments [204]. The use of MLOps enables automation, versioning, reproducibility, etc., with successful collaboration of required skills such as data engineer, data scientist, ML engineer/developer [55] [195]. For example,

data scientists must specialize in SE skills such as modularization, testing, versioning, etc. [206]. Supporting processes formalized in policies serve as the basis for governance [199] and can be automated to ensure solution reliability and compliance [199]. MLOps also support explainability (GDPR regulation [123]) and audit trails [55].

9.5 MLOps Framework and Maturity MOdel

Based on the SLR and the GLR, we derive an MLOps framework that identifies the activities involved in MLOps adoption. Figure 2 depicts the MLOps framework. The entire framework is divided into three pipelines: a) Data Pipeline b) Modeling Pipeline and c) Release Pipeline. After collecting data relevant to ML models from data sources, preprocessing of data and feature extraction is performed. Once a suitable model has been experimented and optimized with hyperparameters, evaluate the model and package it for production deployment. If performance degrades, trigger retraining of the model by initiating a data feedback loop. Data and code that have been versioned are stored in the data repository and code repository. To track the deployable model version, store it in the model registry. Deployment cycles of ML models can be shortened using CI/CD/CT.

MLOps Maturity Model: Based on the SLR and the GLR, we present a maturity model in which we outline four stages in which companies evolve when adopting MLOps practices. The four stages are a) Automated Data Collection b) Automated Model Deployment c) Semi-automated Model Monitoring and d) Fully-automated Model Monitoring. These stages capture key transition points in the adoption of MLOps in practice. Below, we detail each MLOps stage and preconditions for a company to reach this stage.

A. Automated Data Collection: In this stage, companies have a manual processing of data, model, deployment and monitoring. With the adoption of MLOps, company experience a transition from manual process to automated data collection for (re)training process.

Preconditions: For transition from manual process to automated data collection, there is a need for mechanism to aggregate data from different data sources which can be stored and accessed whenever required [198]. In addition, it also demands capability for integrating and processing new data sources, regardless of variety, volume or velocity [199]. It also requires infrastructure resources for automated data collection [207], data prepara-

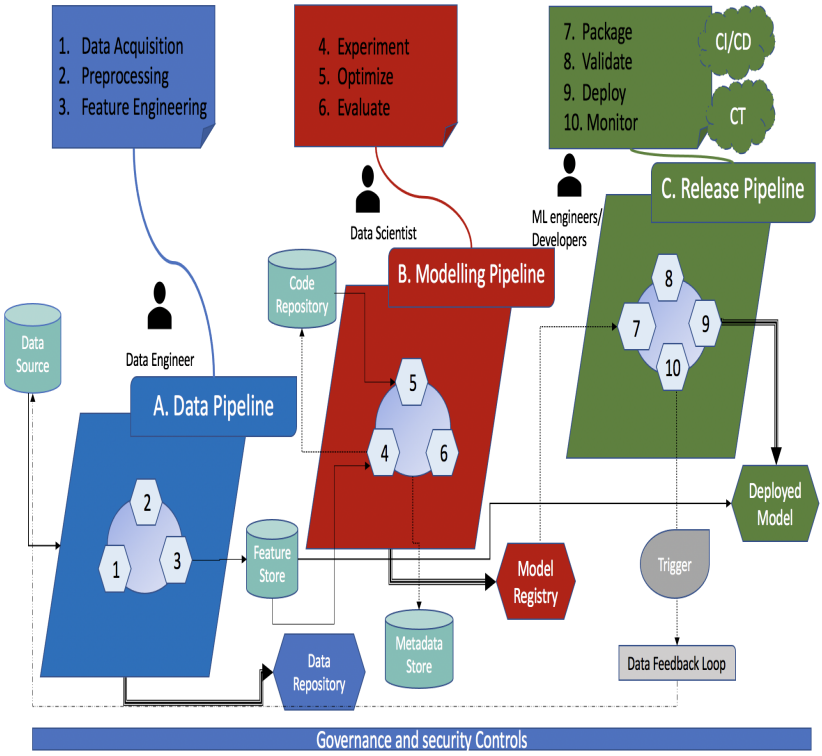

Figure 26: MLOps Framework

tion and collaboration [208]. Also, standardized and automated pipelines helps to drive ingestion, transformation and storage of analytic data into a database or data lake [199]. Same feature manipulation during training has to be replicated at the inference [205]. AI teams can promote trust by addressing data management challenges like accountability, transparency, regulation and compliance, and ethics [209].

B. Automated Model Deployment: The companies at this stage have a manual model deployment and monitoring. With the adoption of MLOps, they undergo transition from manual model deployment and monitoring to automated deployment of the retrained model.

Preconditions: The transition can be achieved by implementing provisions for automated model deployment to environments [196] [203] [196] [208] especially across dev, Q/A and production environments [196] [207]. It encourages deployment freedom in on-premise, cloud and edge [207] [208]. Automated deployment of retrained model can be achieved by provid-

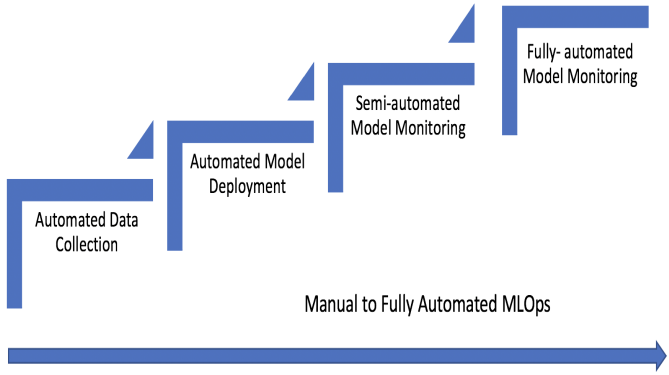

Figure 27: MLOps Maturity Model

ing a dedicated infrastructure-centric CI/CD pipeline [199], integration with DevOps for automation, scale and collaboration [205]. Sufficient infrastructure choices for deployment includes model hosting, evaluation, and maintenance [198], and means to register, package (containerization [208] [210]), deploy models [196] [55] [203] and integration of reusable software environments for training and deploying models [196]. Tracking experiments [196] [203] [55] and models [199], proper validation of models and data [206] can accelerate automated model deployment. Canary Deployments [198] and provisions to store, annotate, discover, and manage models in a central repository [203] can facilitate automated deployment of retrained model. This stage also requires multi-talented teams of technologists and ML professionals to operationalize and scale AI [209].

C. Semi-automated Model Monitoring: At this stage, companies have a manual model monitoring in place. With MLOps, they can attain a transition from manual monitoring to semi-automated model monitoring.

Preconditions: To reach this transition, there should be provisions for triggering [196] when performance degrades and availability of tools for diagnostics, performance monitoring and addressing model drift [196] [201] [206]. It also requires automation scripts to manage and monitor models based on drift [208] and ability to perform continuous model tracking [199]. For easy monitoring of models, MLOps professionals has to be provided with visual tools [207], and dedicated and centralized dashboards [208] [201] [194]. It also requires data orchestration pipelines and rule-based data governance to ensure data changes [199], feedback loop

and continuous model retraining [196]. There should be also a mechanism to automatically train model in production using fresh data based on live pipeline triggers and feedback loops [208]
D. Fully-automated Model Monitoring: The companies have deployment and monitoring of models in place where performance degradation is acknowledged by alert. By utilizing MLOps, they undergo transition towards fully automated monitoring of models.
Preconditions: For this transition, company requires CI/CD integration with automation and orchestration [196] and CT pipeline to retrain models when performance degrades [199]. For this transition, there is a need to ensure certification of models [198] [193], governance and security controls [196] [207] [206], model explainability [196] [206], auditing of model usage [207] [196], reproducible workflow and models [206]. There should be mechanisms to perform end-to-end QA test and performance checks [196]. There should be assurance that data security and privacy requirements are built into data pipelines [199] as well as retrain production models on newer data using the data, algorithms and code used to create the original [207].

9.6 Validation Study

The framework derived from the previous literature was validated in three case companies. Below, we detail how they have tried to introduce and integrate MLOps into their software development systems.

9.6.1 Case company A

Prior to implementing MLOps, the company planned an initial meeting with team members to discuss realistic expectations for MLOps. According to *P2*, practitioners must spend a significant amount of time creating the architecture, communicating, and discussing MLOps in the beginning, the end result is a significant reduction in manual work and end-to-end automation. According to case company A, the primary goal of MLOps is to achieve a) Automation b) Versioning of datasets and models c) Traceability and d) Reproducibility.

> P1: *"We have already implemented something, but not fully, we are still investigating the concept of MLOps. We are trying to see what is in place and what is not and also how to implement*

that are not in place"

When working with data, practitioners need to have a data pipeline in place and provisions to register training data. Before MLOps was implemented, this process was manual in the company. For instance, when practitioners train a model, they read data from log files and ended up using new data each time to train because they did not have access to old data due to the lack of a data pipeline. To ensure the quality of the data pipeline, data schema has to be validated. The company is thinking about using DVC (Data Version Control) to facilitate comparison of different models and visualizations. When dealing with models, practitioners keep track of model performance by tuning them with hyperparameters and maintain the quality of the model pipeline.

Practitioners in case company A place more emphasis on understanding how a project works, especially when it comes to the concept of model deployment. According to P3, the best way to verify that integration of ML code into a project works or not is to have a CI or mechanism to test it without running it in production. For example, practitioners develop a program that can run on a laptop and give everyone access to a local repository and environment. In such a way they can run the data and the entire project with less data set (for instance, 10 percent of the total size of the data set) to make sure it works properly. If the models work properly in the development environment, practitioners will move them to the production environment. To integrate these models, they need to consider other models that are already in production. As a result, it takes time for practitioners to understand how to feed previous data into the system. The company uses Tableau for data visualization and Graffana for model monitoring.

In case company A, data pipeline is quite immature and the model pipeline is not fully automated. On the other hand, model serving pipeline is quite aligned with MLOps. The company utilizes dataset versioning to compare models. They earlier uses Gate for versioning and recently moved to artifactory.

> W5: *"We have versioning for data and model but may be in future we will go for versioning up the entire pipeline, configurations etc"*

> P1: *"There are somethings which we do in our day to day work*

which are continuous and some of these can be automatized. A data scientist should focus on feature engineering, model training etc., instead of deployment or even writing test cases which can be automatized to a great extent"

9.6.2 Case company B

Case company B is also trying to implement MLOps in its context. In dealing with data, the company collects data from real vehicles or generates it using simulations. They input this data into a logging system and add metadata to the logs. After labeling, they ensure the quality of the images. Practitioners access the logged data via an API, which is offloaded from hard drives to servers. To extract data, they perform property selection on selected data frames, run queries to find data to select for future investigation, run algorithms to find valuable data and anonymize data. The company also looks forward to ensuring consistency of annotations throughout the project. Once the dataset is annotated, they perform pre-processing and split the dataset into training, validation, and testing.

When it comes to models, they train neural networks, spin computational nodes, and deallocate them after training. They back up the experiments and validate the models and use model pruning to increase inference speed. The practitioners convert the model using ONXX and deploy it to artifactory. Once the model is deployed in Artifactory, it can be used in vehicles. They essentially deploy the pipeline using the CI/CD loop. They also update the validation set based on new data, domain, etc. The company is interested in moving from on-premise to cloud services for scalability. The company is creating artifacts.

> W8: *"We build artifacts as we need tool chain for data selection, development and deployment on target devices to run inferences. We depend on other teams inside the company for certain artifacts. For instance, the logging system. Besides that, the team build the rest of the artifacts."*

9.6.3 Case company C

Case company C drive towards attaining fully automated MLOps. Company has standardized and quality model development, manageable deploy-

ment workflow and model lifecycle in production. The main architectural principles of case company C is to achieve a) Scale and high availability b) Flexibility and extensibility c) Integration d) Automation and e) Maintainability.

When dealing with data, the company captures images of packages using cameras or generate them using simulations. The data captured is stored in data lake before using it for training. The practitioners experiment with several algorithms before finalizing the best suitable model and adopt hyperparameter tuning. They utilize GPUs for training to reduce needed time. The company applies DevOps principles to their ML systems. The models are packaged and deployed to production via docker containers. They employ Kubernetes to automate deployment as well as scaling. The company has provisions for tracking data, models and experiments. They have model management in place and also models can be deployed to cloud, edge and on-premise. The model monitoring can be visualized using dashboards and retrain models when performance degrades. The company also uses tool chain for development and deployment of models.

9.7 Discussion

This study highlights the emerging interest in MLOps and the increasing adoption of these practices in software-intensive systems. Compared to SLRs ([193] - [194]), more relevant GLR studies [195] - [196] on MLOps are retrieved from the literature. This is a positive sign as it gives an indication that more companies are driving towards achieving fully automated MLOps. Both the SLR and the GLR emphasize the fact that cross-functional teams with skills from data engineering, data science, or operations can facilitate MLOps. Based on insights from the literature, we see that feature store, data repository, code repository, metadata store, model registry, and feed-back loops can shorten the transition of models from prototype to production stage. As a result, they promote automation, versioning, explainability, and traceability.

In Figure 4, we see that case company A is placed in between the phases - Automated Data Collection and Automated Model deployment. This is because while the modeling and deployment pipelines in this company are very mature, the data pipeline is immature. Also the company intends to version the data, model and release pipelines. The challenges faced by case company A in the beginning phases of MLOps corresponds to

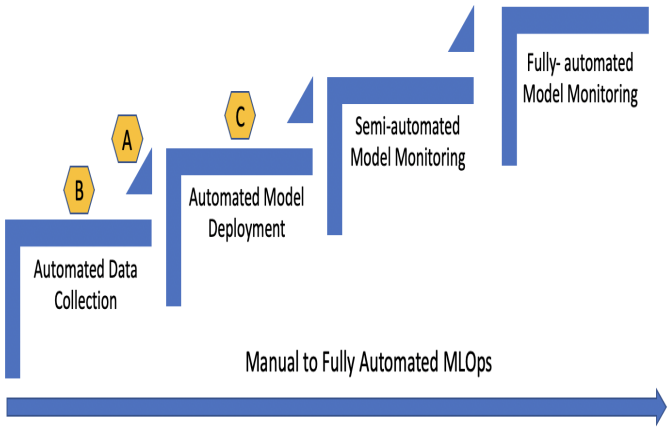

Figure 28: Mapping: Case companies to stages in Maturity model

challenges we identified in literature reviews. The data pipeline challenges that company A is experiencing is something common and also reported by other companies in studies that were part of the GLR. Similar to case company A, we place company B at stage one in the maturity model - Automated Data Collection. This is because they are looking forward to ensuring consistency of annotations across project. On the other hand, they have provisions for data collection from multiple sources, queries and algorithms to run valuable data, experiment tracking, etc. Since their data pipeline is not completely automated, we place them in stage one. Case company C is placed in stage two - Automated Model deployment in the maturity model. They employ DevOps principles in their application, data lakes available for collecting data, frameworks for running experiments, track experiments, utilize docker containers for deployment. They also have mechanisms to deploy models in cloud or edge. Whenever degrades, they initiate model retraining and has model management in place. Even though they have mechanisms for automated model deployment, they are placed in stage two as they look forward to achieve fairness, generalizability, explainability and governance of models.

9.8 Conclusion

Companies adopt DevOps principles to ML systems in order to allow continuous development, deployment and delivery of these systems. In this paper, we derive a framework that identifies the activities involved when

adopting MLOps and the stages in which companies evolve as they become more advanced. We validate this framework in three software-intensive embedded systems companies and highlight how they have managed to adopt and integrate MLOps into their large-scale software development organizations. In future research, we plan to expand our study by involving additional case companies and experts for validation of our results. We believe our findings support successful adoption of MLOps in software-intensive embedded systems companies.

10 Conclusion

To sum up, this research explores the need for systematic and structured design methods and processes for the end-to-end process of developing, deploying and successfully evolving ML/DL models. The main goal of this research is to enable not only experienced data scientists but also less experienced data scientists, software engineers and even non-experts to approach and promote the development, deployment and evolution of ML/DL models. In this thesis, we present the motivation, research methodology and findings of our research in developing an end-to-end process for the development, deployment and evolution of ML/DL models in embedded systems domain companies. In this research, we have mainly focused on the end-to-end process of ML/DL model development as well as in-depth studies focusing on the specifics of the deployment phase and the operationalization of these models in large-scale embedded systems and on the continuous delivery and evolution of Machine Learning Systems (MLOps).

10.1 Answering the RQs

In order to answer the RQs formulated in Chapter 3, we conducted our research in close collaboration with the companies of Software Center, using different empirical research methods such as case studies, action research and literature review as well as different research techniques such as interviews, observations and multi-vocal literature review. We believe that answering these RQs will accelerate the transition from prototypes to production-ready ML/DL models and provide support to practitioners regardless of their experience in working with ML/DL models.

RQ1: What are the activities performed and challenges experienced in the process of developing ML/DL models in companies with software-intensive embedded systems?

To make the development of well-functioning ML/DL models accessible to more than just experienced data scientists, we have developed a design process based on the activities of experienced data scientists in case companies. The seven typical phases that data scientists go through when developing ML/DL models include (i) Business case specification, (ii) Data exploration, (iii) Feature engineering, (iv) Experimentation, (v) Development, (vi) Deployment and (vii) operational. In addition, we have identified iterations between these seven phases and the events that trigger these iterations to optimize the design process. These are: (i) Deployment to Development phase, (ii) Deployment to Experimentation phase, (iii) Development to Experimentation phase, (iv) Experimentation to Feature engineering phase and (v) Development to Data exploration phase. The use of iterations can help build even better models by ensuring better prediction, efficient inference and high business value. We have also identified the key challenges that data scientists experience at each phase of the design process when developing the ML/DL model. In business case specification phase, the challenges are high costs, communication gap, high AI expectations, less data scientists, and large dataset needed. In data exploration phase, the challenges are privacy concerns & noisy data, shortage in domain experts and labeling. In feature engineering phase, they encounter challenges such as increasing complexity and improper feature selection. Challenges in experimentation phase include introduce bias, uncertainty, high complex DL models, need for deep DL knowledge and related work. Challenges in the development phase include determining final model, model execution environment, more hyperparameter settings, verification and validation. In the deployment phase, practitioners deploy less DL models, integration issues, internal deployment and need for an intelligible model. In the operational phase, the main challenges are training-serving Skew, end-user communication, model drifts and maintain robustness.

In an extended study, the identified phases are categorized into three high-level activities (i.e., focusing on business, data and models) that companies perform in parallel to develop, experiment and optimize the models they develop. The activities are business case experimentation, data experimentation and model experimentation. Each activity consists of four stages. In the case of business case experimentation, these stages are as follows: (i) Multiple business case generation, (ii) Qualitative selection of each generated case, (iii) Selection and specification of the prioritized business case, and (iv) Business case validation. Data Experimentation in-

cludes: (i) Data generation, (ii) Data collection, (iii) Data exploration, and (iv) Operational. Model experimentation has (i) Feature engineering, (ii) Experimentation, (iii) Development and (iv) Deployment. The previously identified critical challenges are grouped into three categories related to the development of ML/DL models: a) Pre-deployment, b) Deployment, and c) Non-technical challenges. We also identified additional events that trigger iterations. These are: (i) Business case validation to Selection and specification of prioritized business case, (ii) Experimentation to Data collection, (iii) Data exploration to Selection and specification of prioritized business case and (iv) Development to Data collection. Different profiles of practitioners involved in the development, experimentation and optimization of ML/DL models are business owners, product owners, UX designers, data practitioner, domain experts, developers, solution architects and front-end/back-end developers. We have identified several checkpoints for business case rejection (i) It does not make sense or provide value in the early stages of business case experimentation, (ii) Innovation management does not find business value for the case, (iii) It fails to find a business owner who sees value in the case, (iv) There is no existing data set, (v) Proof of concept never justifies model development (vi) Expensive and difficult data generation and data collection methods, (vii) It is not cost-effective or approved by business owners during business case validation.

RQ2:What are the best practices, challenges encountered and architectural choices made by practitioners in deploying and operationalizing ML/DL models?
For the deployment of AI at the edge, we have developed a generic framework consisting of five architectural alternatives, ranging from a centralized architecture where (re)training in the cloud takes precedence to a decentralized architecture where (re)training at the edge takes precedence instead. In Architecture 1 - Global optimum approach: the traditional ML/DL systems tend towards a centralized approach where the global model is (re)trained in the cloud and a child model is put into operation locally at the edge. In architecture 2 - Global optimum approach + Local data for actual global retraining: the global model (re)training takes place in the cloud and the local operation is performed at the edge. In this architecture, the mislabeled data identified after inference by human intervention is returned to the cloud. In architecture 3 - Local optimum approach + Transfer Learning: the (re)training of the global model takes place in the

cloud and the (re)training of the child model and integration with the local operation is done at the edge. In architecture 4 - Local optimum approach + Federated learning + Local data for actual retraining: the (re)training of the global model takes place in the cloud and the (re)training of the child model and placement in the local operation takes place at the edge. In architecture 5 - Local optimum approach + Diverging deployment: the (re)training of the global model takes place in the cloud and the (re)training of the child model and the placement in the local operation takes place at the edge. We have also presented two variants on the architectural alternatives identified in the framework. These are a) Quorum of child models and b) Wrapper around the global model. Finally, we have identified the main challenges practitioners face in selecting an ideal architecture alternative. The challenges include strong internet connection, hardware requirements, shortage of data scientists, expensive cloud solutions, lack of edge computing power, understand KPIs to achieve, dataset copyrights, need massive dataset, find relevant dataset for model, corner cases to improve generalization, local training dataset prone to domain shifts, domain experts shortage and human involvement in labeling. low performance of the pretrained models for transfer learning, model fails to do generalization, general-purpose platform to easily change algorithms, failures during global model deployment at edge, difficulty in getting labeled data once deployed to edge, scheduling the retraining interval, traceability to find misbehaving models, and robustness.

In a follow-up study to the generic framework for deploying ML/DL models, we identified factors that help practitioners decide which architecture to choose for deploying ML/DL models, i.e., device cost, model performance, and privacy. We have also developed a framework outlining how prioritization and trade-offs between these factors lead to a particular architecture. These are (i) Cost and Performance, (ii) Cost and Privacy, and (iii) Performance and Privacy. We have also discussed additional factors that may or may not influence the selection of an optimal architecture. These are device architecture and the kind of computation available in the device architecture, explainability and scalability.

To improve the understanding of AI model integration, deployment and operationalization, we derived a framework that accelerates the end-to-end deployment process. The framework is divided into five phases: Design, Integration, Deployment, Operation, and Evolution. Each phase consists of two tasks. These are: (a) Design - Validation & Tracking (b) Integration

- Resource discovery & Rewrite/Package (c) Deployment - Target environment & Launching (d) Operation - Inference & Monitoring (e) Evolution - Retrain & Redeploy.

RQ3: How can MLOps practices help large-scale embedded systems companies achieve continuous delivery and evolution of ML/DL models?
To ensure the continuous delivery of ML/DL models, we have developed a conceptual framework to facilitate AI-driven business development. With this framework, we have provided practitioners with a blueprint for effectively incorporating AI/ML/DL into a company's business in the context of a larger system. We have also derived a framework that details the activities involved in the continuous development of ML models. The framework is divided into three pipelines: a) Data pipeline b) Modeling pipeline and c) Release pipeline. We store versioned data and code in the data repository and code repository respectively. For tracking purposes, the deployed model version is stored in the model registry. Moreover, CI/CD/CT (Continuous Integration /Continuous Deployment/ Continuous Training) can shorten the deployment cycles of ML models. Finally, we have developed a maturity model in which we outline four stages that companies go through when adopting MLOps practices. MLOps is a practice that brings together data scientist teams and operations in companies. The four stages are a) Automated data collection b) Automated model deployment c) Semi-automated model monitoring and d) Fully-automated model monitoring. These stages are mapped to three case companies in the embedded systems domain.

Figure 29 shows all the frameworks developed in the research to facilitate the end-to-end process of ML/DL model development, deployment and evolution.

10.2 Key Contributions

Based on our longitudinal case study with companies in software-intensive embedded systems domain at Software Center from August 2019 to June 2021, we present an end-to-end process of developing, deploying, and evolving ML/DL models as the main contribution. For the development of ML/DL models, we derived a design process for the development of ML/DL models along with iterations to optimize the design process as well as associated challenges. For the deployment of AI models in the cloud,

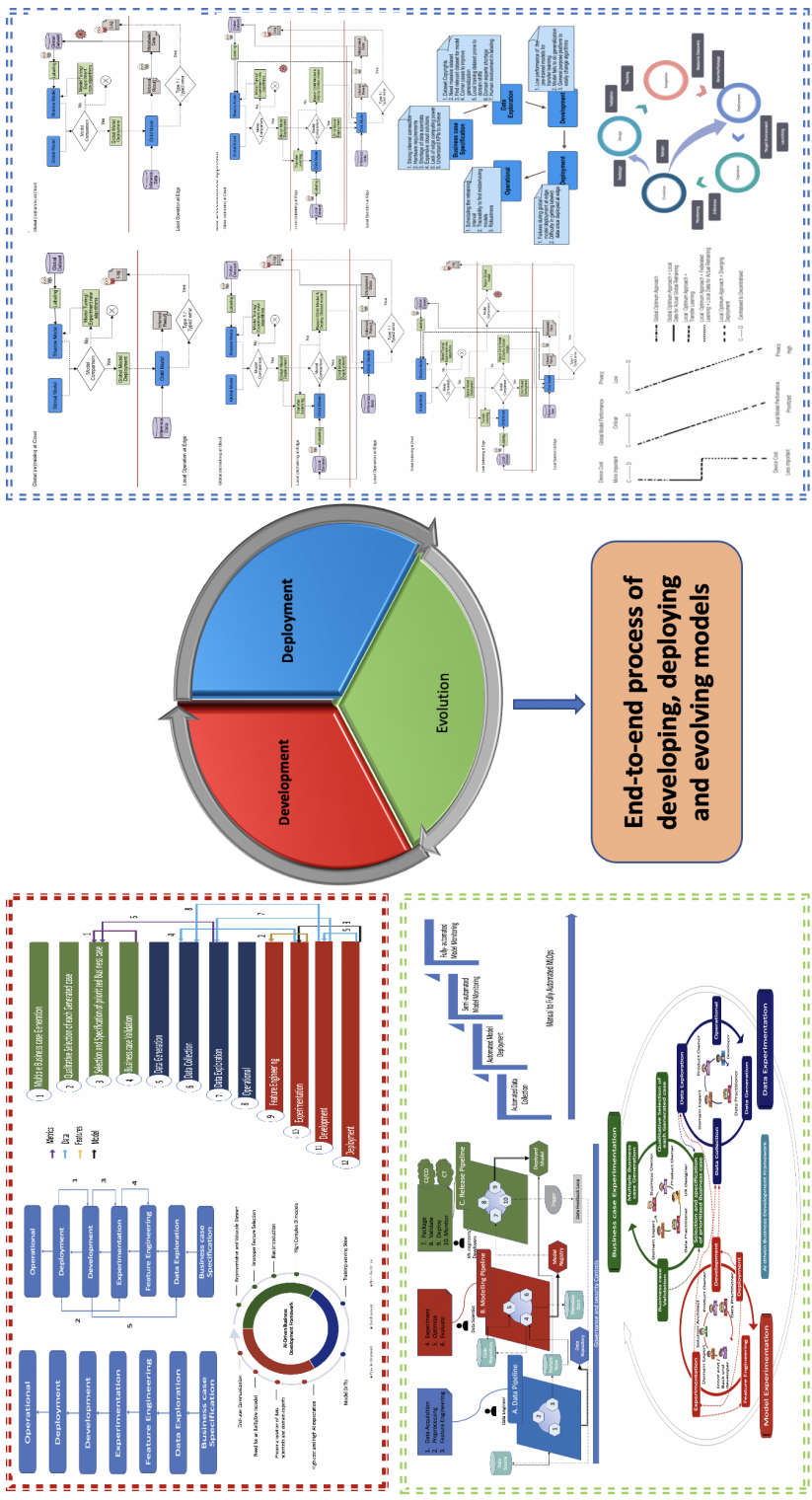

Figure 29: End-to-end process of Developing, Deployment and Evolving ML/DL Models

edge or a mix of cloud and edge, we have developed a generic framework with five architectural alternatives for the deployment of AI models. We then identify key factors, as well as priorities and trade-offs among key factors that are critical to the deployment of AI systems. To accelerate the deployment and integration of ML/DL models, we have developed an end-to-end deployment framework. To ensure the continuous deployment and evolution of ML/DL models, we developed an MLOps framework and proposed an AI-driven Business Development Framework. The main contributions are as follows:

- Design process for development of ML and DL models with iterations to optimize the design process
- Development of a generic framework for deploying AI at the edge
- Architecture selection framework for AI system deployment
- Development of End-to-end Deployment Framework
- Development of MLOps Framework
- Proposed AI-driven Business Development Framework
- Identification of key challenges that practitioners face when developing ,deploying and evolving ML and DL models

Combining ML and DL with embedded systems in companies provide the ability to collect data, analyse, and predict. This process can improve the performance of embedded systems and provide smarter solutions. The research goals are achieved through the studied cases in various domains such as telecommunication, automotive, packaging, manufacturing, etc. Some of the work in one company may be identified differently in parts of other companies. We do not claim that the opportunities and challenges are the same for different industries and disciplines.

10.3 Future work

Evaluating architectural alternatives for deploying AI on the edge based on the key factors involves experimenting with three architectural alternatives. These are (a) Centralized architecture (b) Federated architecture and (c) Decentralized architecture. These are validated against key factors i.e.,

Mass customization, Scalability and Model Performance. This is a follow-up study to the generic framework for deploying ML/DL models developed as part of the action research in one of the SC companies and validated in other SC companies in previous sprints.

References

[1] "CRediT author statement elsevier." https://www.elsevier.com/authors/journalauthors/policies-and-ethics/credit-author-statement. Accessed: 2021-06-18.

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

[3] G. Giray, "A software engineering perspective on engineering machine learning systems: State of the art and challenges," *arXiv preprint arXiv:2012.07919*, 2020.

[4] M. Bojarski, D. Del Testa, D. Dworakowski, B. Firner, B. Flepp, P. Goyal, L. D. Jackel, M. Monfort, U. Muller, J. Zhang, *et al.*, "End to end learning for self-driving cars," *arXiv preprint arXiv:1604.07316*, 2016.

[5] M. F. Dixon, I. Halperin, and P. Bilokon, *Machine Learning in Finance*. Springer, 2020.

[6] J. B. Heaton, N. G. Polson, and J. H. Witte, "Deep learning for finance: deep portfolios," *Applied Stochastic Models in Business and Industry*, vol. 33, no. 1, pp. 3–12, 2017.

[7] V. Gulshan, L. Peng, M. Coram, M. C. Stumpe, D. Wu, A. Narayanaswamy, S. Venugopalan, K. Widner, T. Madams, J. Cuadros, *et al.*, "Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs," *Jama*, vol. 316, no. 22, pp. 2402–2410, 2016.

[8] T. Y. Wong and N. M. Bressler, "Artificial intelligence with deep learning technology looks into diabetic retinopathy screening," *Jama*, vol. 316, no. 22, pp. 2366–2367, 2016.

[9] 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)*, pp. 50–59, IEEE, 2018.

[10] 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*, pp. 227–243, Springer, Cham, 2019.

[11] 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)*, pp. 140–147, IEEE, 2019.

[12] "Gartner Identifies the Top Strategic Technology Trends for 2021." https://www.gartner.com/en/newsroom/press-releases/2020-10-19-gartner-identifies-the-top-strategic-technology-trends-for-2021. Accessed: 2021-07-01.

[13] 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)*, pp. 162–170, IEEE, 2016.

[14] D. Sculley, G. Holt, D. Golovin, E. Davydov, T. Phillips, D. Ebner, V. Chaudhary, and M. Young, "Machine learning: The high interest credit card of technical debt," 2014.

[15] 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*, pp. 2503–2511, 2015.

[16] L. Baier, F. Jöhren, and S. Seebacher, "Challenges in the deployment and operation of machine learning in practice," 2019.

[17] S. Wang, L. Huang, J. Ge, T. Zhang, H. Feng, M. Li, H. Zhang, and V. Ng, "Synergy between machine/deep learning and software engineering: How far are we?," *arXiv preprint arXiv:2008.05515*, 2020.

[18] 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)*, pp. 291–300, IEEE, 2019.

[19] C. Murphy, G. E. Kaiser, and M. Arias, "An approach to software testing of machine learning applications," 2007.

[20] 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*, pp. 2059–2068, 2015.

[21] Q. Yang, J. Suh, N.-C. Chen, and G. Ramos, "Grounding interactive machine learning tool design in how non-experts actually build models," in *Proceedings of the 2018 Designing Interactive Systems Conference*, pp. 573–584, 2018.

[22] D. Wang, J. D. Weisz, M. Muller, P. Ram, W. Geyer, C. Dugan, Y. Tausczik, H. Samulowitz, and A. Gray, "Human-ai collaboration in data science: Exploring data scientists' perceptions of automated ai," *Proceedings of the ACM on Human-Computer Interaction*, vol. 3, no. CSCW, pp. 1–24, 2019.

[23] S. Miller and D. Hughes, "The quant crunch: How the demand for data science skills is disrupting the job market," *Burning Glass Technologies*, 2017.

[24] S. Miller, "Collaborative approaches needed to close the big data skills gap," *Journal of Organization design*, vol. 3, no. 1, pp. 26–30, 2014.

[25] "2017 Kaggle Machine Learning Data Science Surveyb." `https://www.kaggle.com/kaggle/kaggle-survey-2017/`. Accessed: 2021-07-07.

[26] S. S. Nazrul, "Devops for data scientists: Taming the unicorn," *Medium: Towards Data Science*, 2018.

[27] W. M. Van der Aalst, "Data scientist: The engineer of the future," in *Enterprise interoperability VI*, pp. 13–26, Springer, 2014.

[28] M. Muller, I. Lange, D. Wang, D. Piorkowski, J. Tsay, Q. V. Liao, C. Dugan, and T. Erickson, "How data science workers work with data: Discovery, capture, curation, design, creation," in *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*, pp. 1–15, 2019.

[29] C. Bopp, E. Harmon, and A. Voida, "Disempowered by data: Nonprofits, social enterprises, and the consequences of data-driven work," in *Proceedings of the 2017 CHI conference on human factors in computing systems*, pp. 3608–3619, 2017.

[30] S. Kim, D. Tasse, and A. K. Dey, "Making machine-learning applications for time-series sensor data graphical and interactive," *ACM Transactions on Interactive Intelligent Systems (TiiS)*, vol. 7, no. 2, pp. 1–30, 2017.

[31] J. Bosch, H. H. Olsson, and I. Crnkovic, "It takes three to tango: Requirement, outcome/data, and ai driven development.," 2018.

[32] "Digitalization." `https://www.gartner.com/en/information-technology/glossary/digitalization`. Accessed: 2021-07-06.

[33] F. Giaimo and C. Berger, "Design criteria to architect continuous experimentation for self-driving vehicles," in *2017 IEEE International Conference on Software Architecture (ICSA)*, pp. 203–210, IEEE, 2017.

[34] J. Bosch and U. Eklund, "Eternal embedded software: Towards innovation experiment systems," in *International Symposium On Leveraging Applications of Formal Methods, Verification and Validation*, pp. 19–31, Springer, 2012.

[35] K. Pohl, *Requirements engineering: fundamentals, principles, and techniques*. Springer Publishing Company, Incorporated, 2010.

[36] F. Paetsch, A. Eberlein, and F. Maurer, "Requirements engineering and agile software development," in *WET ICE 2003. Proceedings. Twelfth IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises, 2003.*, pp. 308–313, IEEE, 2003.

[37] H. H. Olsson and J. Bosch, "Towards evidence-based development: learnings from embedded systems, online games and internet of things," *IEEE Software*, vol. 4, no. 5, 2017.

[38] 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*, pp. 9–16, IEEE, 2014.

[39] H. H. Olsson and J. Bosch, "Towards data-driven product development: A multiple case study on post-deployment data usage in software-intensive embedded systems," in *International Conference on Lean Enterprise Software and Systems*, pp. 152–164, Springer, 2013.

[40] 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*, vol. 33, no. 5, p. e2333, 2021.

[41] "DevOps." `https://devops.com/`. Accessed: 2021-07-06.

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

[43] 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*, pp. 165–174, 2020.

[44] S. Mäkinen, H. Skogström, E. Laaksonen, and T. Mikkonen, "Who needs mlops: What data scientists seek to accomplish and how can mlops help?," *arXiv preprint arXiv:2103.08942*, 2021.

[45] "MLOps: Continuous delivery and automation pipelines in machine learning." https://cloud.google.com/architecture/mlops-continuous-delivery-and-automation-pipelines- in-machine-learning. Accessed: 2021-07-06.

[46] A. Goyal, "Machine learning operations," *International Journal of Information Technology Insights & Transformations [ISSN: 2581-5172 (online)]*, vol. 4, no. 2, 2020.

[47] J. Soh and P. Singh, "Machine learning operations," in *Data Science Solutions on Azure*, pp. 259–279, Springer, 2020.

[48] A. L. Samuel, "Some studies in machine learning using the game of checkers," *IBM Journal of research and development*, vol. 44, no. 1.2, pp. 206–226, 2000.

[49] T. M. Mitchell, "Does machine learning really work?," *AI magazine*, vol. 18, no. 3, pp. 11–11, 1997.

[50] J. Alzubi, A. Nayyar, and A. Kumar, "Machine learning from theory to algorithms: an overview," in *Journal of physics: conference series*, vol. 1142, p. 012012, IOP Publishing, 2018.

[51] S. Das, A. Dey, A. Pal, and N. Roy, "Applications of artificial intelligence in machine learning: review and prospect," *International Journal of Computer Applications*, vol. 115, no. 9, 2015.

[52] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," in *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 17–36, JMLR Workshop and Conference Proceedings, 2012.

[53] D. Xin, L. Ma, S. Song, and A. Parameswaran, "How developers iterate on machine learning workflows–a survey of the applied machine learning literature," *arXiv preprint arXiv:1803.10311*, 2018.

[54] M. Salvaris, D. Dean, and W. H. Tok, "Microsoft ai platform," in *Deep Learning with Azure*, pp. 79–98, Springer, 2018.

[55] "Machine learning workflow." `https://cloud.google.com/ai-platform/docs/ml-solutions-overview`. Accessed: 2021-07-06.

[56] K. Patel, J. Fogarty, J. A. Landay, and B. Harrison, "Investigating statistical machine learning as a tool for software development," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pp. 667–676, 2008.

[57] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth, "The kdd process for extracting useful knowledge from volumes of data," *Communications of the ACM*, vol. 39, no. 11, pp. 27–34, 1996.

[58] R. Wirth and J. Hipp, "Crisp-dm: Towards a standard process model for data mining," in *Proceedings of the 4th international conference on the practical applications of knowledge discovery and data mining*, pp. 29–39, Springer-Verlag London, UK, 2000.

[59] U. Kanewala and J. M. Bieman, "Testing scientific software: A systematic literature review," *Information and software technology*, vol. 56, no. 10, pp. 1219–1232, 2014.

[60] E. Breck, S. Cai, E. Nielsen, M. Salib, and D. Sculley, "What's your ml test score? a rubric for ml production systems," 2016.

[61] J. Dean, G. Corrado, R. Monga, K. Chen, M. Devin, M. Mao, M. Ranzato, A. Senior, P. Tucker, K. Yang, *et al.*, "Large scale distributed deep networks," in *Advances in neural information processing systems*, pp. 1223–1231, 2012.

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

[63] J. K. Nayak and P. Singh, *Fundamentals of Research Methodology Problems and Prospects*. SSDN Publishers & Distributors, 2021.

[64] C. Pope and N. Mays, "Qualitative research: reaching the parts other methods cannot reach: an introduction to qualitative methods in health and health services research," *bmj*, vol. 311, no. 6996, pp. 42–45, 1995.

[65] "Software Centerb." `https://www.software-center.se/`. Accessed: 2021-06-18.

[66] N. K. Denzin and Y. S. Lincoln, "Introduction: The discipline and practice of qualitative research.," 2005.

[67] C. Anderson, "Presenting and evaluating qualitative research," *American journal of pharmaceutical education*, vol. 74, no. 8, 2010.

[68] M. Hennink, I. Hutter, and A. Bailey, *Qualitative research methods*. Sage, 2020.

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

[70] M. Q. Patton, "Qualitative research," *Encyclopedia of statistics in behavioral science*, 2005.

[71] R. K. Yin, "Case study research: Design and methods fourth edition," *Los Angeles and London: SAGE*, 2009.

[72] P. Runeson, M. Host, A. Rainer, and B. Regnell, *Case study research in software engineering: Guidelines and examples*. John Wiley & Sons, 2012.

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

[74] M. Host and P. Runeson, "Checklists for software engineering case study research," in *First international symposium on empirical software engineering and measurement (ESEM 2007)*, pp. 479–481, IEEE, 2007.

[75] M. Staron, *Action Research in Software Engineering*. Springer, 2020.

[76] D. I. Sjoberg, T. Dyba, and M. Jorgensen, "The future of empirical methods in software engineering research," in *Future of Software Engineering (FOSE'07)*, pp. 358–378, IEEE, 2007.

[77] D. E. Avison, F. Lau, M. D. Myers, and P. A. Nielsen, "Action research," *Communications of the ACM*, vol. 42, no. 1, pp. 94–97, 1999.

[78] G. I. Susman and R. D. Evered, "An assessment of the scientific merits of action research," *Administrative science quarterly*, pp. 582–603, 1978.

[79] E. R. Babbie, *The practice of social research*. Cengage learning, 2020.

[80] R. L. Baskerville, "Investigating information systems with action research," *Communications of the association for information systems*, vol. 2, no. 1, p. 19, 1999.

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

[82] R. Davison, M. G. Martinsons, and N. Kock, "Principles of canonical action research," *Information systems journal*, vol. 14, no. 1, pp. 65–86, 2004.

[83] P. Runeson and M. Höst, "Guidelines for conducting and reporting case study research in software engineering," *Empirical software engineering*, vol. 14, no. 2, pp. 131–164, 2009.

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

[85] S. E. Hove and B. Anda, "Experiences from conducting semi-structured interviews in empirical software engineering research," in *11th IEEE International Software Metrics Symposium (METRICS'05)*, pp. 10–pp, IEEE, 2005.

[86] S. Kvale, *Interviews: An introduction to qualitative research interviewing*. Sage Publications, Inc, 1994.

[87] D. L. Morgan, *Focus groups as qualitative research*, vol. 16. Sage publications, 1996.

[88] H. J. Rubin and I. S. Rubin, *Qualitative interviewing: The art of hearing data*. sage, 2011.

[89] F. Shull, J. Carver, and G. H. Travassos, "An empirical methodology for introducing software processes," *ACM SIGSOFT Software Engineering Notes*, vol. 26, no. 5, pp. 288–296, 2001.

[90] M. Shaw, "What makes good research in software engineering?," *International Journal on Software Tools for Technology Transfer*, vol. 4, no. 1, pp. 1–7, 2002.

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

[92] 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.

[93] 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.

[94] 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.

[95] 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.

[96] 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.

[97] J. A. Holton, "The coding process and its challenges," *The Sage handbook of grounded theory*, no. Part III, pp. 265–89, 2007.

[98] V. Wilson, "Research methods: triangulation," *Evidence based library and information practice*, vol. 9, no. 1, pp. 74–75, 2014.

[99] R. E. Stake, *The art of case study research*. sage, 1995.

[100] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[101] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[102] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural networks*, vol. 61, pp. 85–117, 2015.

[103] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, pp. 1097–1105, 2012.

[104] A. Radford, L. Metz, and S. Chintala, "Unsupervised representation learning with deep convolutional generative adversarial networks," *arXiv preprint arXiv:1511.06434*, 2015.

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

[106] A. Efrati, "How deep learning works at apple, beyond," 2017.

[107] K. Hazelwood, S. Bird, D. Brooks, S. Chintala, U. Diril, D. Dzhulgakov, M. Fawzy, B. Jia, Y. Jia, A. Kalro, *et al.*, "Applied machine learning at facebook: A datacenter infrastructure perspective," in *2018 IEEE International Symposium on High Performance Computer Architecture (HPCA)*, pp. 620–629, IEEE, 2018.

[108] Y. Bengio and S. Bengio, "Modeling high-dimensional discrete data with multi-layer neural networks," in *Advances in Neural Information Processing Systems*, pp. 400–406, 2000.

[109] K. Patel, J. Fogarty, J. A. Landay, and B. L. Harrison, "Examining difficulties software developers encounter in the adoption of statistical machine learning.," in *AAAI*, pp. 1563–1566, 2008.

[110] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the people: The role of humans in interactive machine learning," *Ai Magazine*, vol. 35, no. 4, pp. 105–120, 2014.

[111] R. Ashmore, R. Calinescu, and C. Paterson, "Assuring the machine learning lifecycle: Desiderata, methods, and challenges," *arXiv preprint arXiv:1905.04223*, 2019.

[112] G. Walsham, "Interpretive case studies in is research: nature and method," *European Journal of information systems*, vol. 4, no. 2, pp. 74–81, 1995.

[113] R. K. Yin, "Case study research: design and methods (ed.)," *Thousand Oaks*, 2003.

[114] H. H. Olsson and J. Bosch, "Going digital: Disruption and transformation in software-intensive embedded systems ecosystems," *Journal of Software: Evolution and Process*, vol. 32, no. 6, p. e2249, 2020.

[115] " Machine learning workflow b." https://cloud.google.com/ai-platform/docs/ml-solutions-overview,. Accessed: 2021-07-01.

[116] M. M. John, H. H. Olsson, and J. Bosch, "Developing ml/dl models: A design framework," in *Proceedings of the International Conference on Software and System Processes*, pp. 1–10, 2020.

[117] "Digitalizationb." https://www.gartner.com/en/information-technology/glossary/digitalization. Accessed: 2021-03-16.

[118] F. Erich, C. Amrit, and M. Daneva, "A mapping study on cooperation between information system development and operations," in *International Conference on Product-Focused Software Process Improvement*, pp. 277–280, Springer, 2014.

[119] L. E. Lwakatare, T. Kilamo, T. Karvonen, T. Sauvola, V. Heikkilä, J. Itkonen, P. Kuvaja, T. Mikkonen, M. Oivo, and C. Lassenius, "Devops in practice: A multiple case study of five companies," *Information and Software Technology*, vol. 114, pp. 217–230, 2019.

[120] D. Stahl, T. Martensson, and J. Bosch, "Continuous practices and devops: beyond the buzz, what does it all mean?," in *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pp. 440–448, IEEE, 2017.

[121] S. Alla and S. K. Adari, "Beginning mlops with mlflow,"

[122] R. Penners and A. Dyck, "Release engineering vs. devops-an approach to define both terms," *Full-scale Software Engineering*, pp. 49–54, 2015.

[123] D. A. Tamburri, "Sustainable mlops: Trends and challenges," in *2020 22nd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, pp. 17–23, IEEE, 2020.

[124] "Demystified: AI, Machine Learning, Deep Learningb." https://towardsdatascience.com/demystified-ai-machine-learning-deep-learning-c5259d38678e. Accessed: 2021-03-16.

[125] "Artificial Intelligence Industry 4.0: 5 Manufacturing Applications for AIb." https://acerta.ai/blog/artificial-intelligence-industry-4-0-5-manufacturing-applications-for-ai/. Accessed: 2021-03-16.

[126] D. Dahlmeier, "On the challenges of translating nlp research into commercial products," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 92–96, 2017.

[127] R. Rana, M. Staron, J. Hansson, M. Nilsson, and W. Meding, "A framework for adoption of machine learning in industry for software defect prediction," in *2014 9th International Conference on Software Engineering and Applications (ICSOFT-EA)*, pp. 383–392, IEEE, 2014.

[128] R. Hirt, N. J. Koehl, and G. Satzger, "An end-to-end process model for supervised machine learning classification: from problem to deployment in information systems," in *Designing the Digital Transformation: DESRIST 2017 Research in Progress Proceedings of the 12th International Conference on Design Science Research in Information Systems and Technology. Karlsruhe, Germany. 30 May-1 Jun.*, pp. 55–63, Karlsruher Institut für Technologie (KIT), 2017.

[129] K. Bierzynski, A. Escobar, and M. Eberl, "Cloud, fog and edge: Cooperation for the future?," in *2017 Second International Conference on Fog and Mobile Edge Computing (FMEC)*, pp. 62–67, IEEE, 2017.

[130] H.-S. Kim, "Fog computing and the internet of things: Extend the cloud to where the things are," *International Journal of Cisco*, 2016.

[131] H. Li, K. Ota, and M. Dong, "Learning iot in edge: Deep learning for the internet of things with edge computing," *IEEE network*, vol. 32, no. 1, pp. 96–101, 2018.

[132] T. X. Tran, A. Hajisami, P. Pandey, and D. Pompili, "Collaborative mobile edge computing in 5g networks: New paradigms, scenarios, and challenges," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 54–61, 2017.

[133] N. Talagala, S. Sundararaman, V. Sridhar, D. Arteaga, Q. Luo, S. Subramanian, S. Ghanta, L. Khermosh, and D. Roselli,

"{ECO}: Harmonizing edge and cloud with ml/dl orchestration," in *{USENIX} Workshop on Hot Topics in Edge Computing (HotEdge 18)*, 2018.

[134] B. Sena, A. P. Allian, and E. Y. Nakagawa, "Characterizing big data software architectures: a systematic mapping study," in *Proceedings of the 11th Brazilian Symposium on Software Components, Architectures, and Reuse*, pp. 1–10, 2017.

[135] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[136] K. Yang, T. Jiang, Y. Shi, and Z. Ding, "Federated learning via over-the-air computation," *IEEE Transactions on Wireless Communications*, vol. 19, no. 3, pp. 2022–2035, 2020.

[137] J. Pan and J. McElhannon, "Future edge cloud and edge computing for internet of things applications," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 439–449, 2017.

[138] " Edge Computing and Machine Learning: Two Sides of the Same Coinb." `https://blog.swim.ai/2017/edge-computing-and-machine-learning`. Accessed: 2021-03-16.

[139] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[140] M. Volk, S. Bosse, D. Bischoff, and K. Turowski, "Decision-support for selecting big data reference architectures," in *International Conference on Business Information Systems*, pp. 3–17, Springer, 2019.

[141] Y. Q. Dai Wenyuan, X. Guirong, and Y. Yong, "Boosting for transfer learning," in *Proceedings of the 24th International Conference on Machine Learning, Corvallis, USA*, pp. 193–200, 2007.

[142] K. Weiss, T. M. Khoshgoftaar, and D. Wang, "A survey of transfer learning," *Journal of Big data*, vol. 3, no. 1, pp. 1–40, 2016.

[143] J. McNiff, *You and your action research project*. Routledge, 2016.

[144] R. L. Baskerville and A. T. Wood-Harper, "A critical perspective on action research as a method for information systems research," *Journal of information Technology*, vol. 11, no. 3, pp. 235–246, 1996.

[145] E.-M. Ihantola and L.-A. Kihn, "Threats to validity and reliability in mixed methods accounting research," *Qualitative Research in Accounting & Management*, 2011.

[146] L. Bernardi, T. Mavridis, and P. Estevez, "150 successful machine learning models: 6 lessons learned at booking. com," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pp. 1743–1751, 2019.

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

[148] M. M. John, H. H. Olsson, and J. Bosch, "Ai on the edge: Architectural alternatives," in *2020 46th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, pp. 21–28, IEEE, 2020.

[149] S. Lucci and D. Kopec, *Artificial intelligence in the 21st century*. Stylus Publishing, LLC, 2015.

[150] B. Kanagal and S. Tata, "Recommendations for all: Solving thousands of recommendation problems daily," in *2018 IEEE 34th International Conference on Data Engineering (ICDE)*, pp. 1404–1413, IEEE, 2018.

[151] S. Wang, C. Liu, X. Gao, H. Qu, and W. Xu, "Session-based fraud detection in online e-commerce transactions using recurrent neural networks," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 241–252, Springer, 2017.

[152] J. A. Maxwell, *Qualitative research design: An interactive approach*, vol. 41. Sage publications, 2012.

[153] D. Crankshaw, J. Gonzalez, and P. Bailis, "Research for practice: prediction-serving systems," *Communications of the ACM*, vol. 61, no. 8, pp. 45–49, 2018.

[154] F. Provost and R. Kohavi, "Guest editors' introduction: On applied research in machine learning," *Machine learning*, vol. 30, no. 2, pp. 127–132, 1998.

[155] S. Keele *et al.*, "Guidelines for performing systematic literature reviews in software engineering," tech. rep., Citeseer, 2007.

[156] B. A. Kitchenham, D. Budgen, and P. Brereton, *Evidence-based software engineering and systematic reviews*, vol. 4. CRC press, 2015.

[157] 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*, pp. 1–6, 2016.

[158] A. Williams, "Using reasoning markers to select the more rigorous software practitioners' online content when searching for grey literature," in *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018*, pp. 46–56, 2018.

[159] X. Xiaojing and S. S. Govardhan, "A service mesh-based load balancing and task scheduling system for deep learning applications," in *2020 20th IEEE/ACM International Symposium on Cluster, Cloud and Internet Computing (CCGRID)*, pp. 843–849, IEEE, 2020.

[160] Y. Li, Z. Han, Q. Zhang, Z. Li, and H. Tan, "Automating cloud deployment for deep learning inference of real-time online services," in *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pp. 1668–1677, IEEE, 2020.

[161] "Data blueprint - IOA Knowledge Baseb." .

[162] "Move the Algorithms, Not the Datab." `https://www.oracle.com/technetwork/database/options/advanced-analytics\/oml19c-white-paper-5601561.pdf`.

[163] "Machine Learning Based Advanced Analytics using Intel Technology."

https://media.bitpipe.com/io_14x/io_147787/item_1954603/machine-learning-based-advanced-analytics-using-intel-ForDistribution-2.pdf.

[164] J. Li, J. Li, and H. Zhang, "Deep learning based parking prediction on cloud platform," in *2018 4th International Conference on Big Data Computing and Communications (BIGCOM)*, pp. 132–137, IEEE, 2018.

[165] "Getting Started with Machine Learning in the Cloud b." `https://www.oracle.com/a/ocom/docs/machine-learning-goes-to-the-cloud-ebook.pdf`.

[166] E. Brumbaugh, M. Bhushan, A. Cheong, M. G.-Q. Du, J. Feng, N. Handel, A. Hoh, J. Hone, B. Hunter, A. Kale, *et al.*, "Bighead: a framework-agnostic, end-to-end machine learning platform," in *2019 IEEE International Conference on Data Science and Advanced Analytics (DSAA)*, pp. 551–560, IEEE, 2019.

[167] S. Serebryakov, D. Milojicic, N. Vassilieva, S. Fleischman, and R. D. Clark, "Deep learning cookbook: recipes for your ai infrastructure and applications," in *2019 IEEE International Conference on Rebooting Computing (ICRC)*, pp. 1–9, IEEE, 2019.

[168] P. Salza, E. Hemberg, F. Ferrucci, and U.-M. O'Reilly, "Towards evolutionary machine learning comparison, competition, and collaboration with a multi-cloud platform," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, pp. 1263–1270, 2017.

[169] M. M. Rovnyagin, K. V. Timofeev, A. A. Elenkin, and V. A. Shipugin, "Cloud computing architecture for high-volume ml-based solutions," in *2019 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (EIConRus)*, pp. 315–318, IEEE, 2019.

[170] N. Gupta, K. Anantharaj, and K. Subramani, "Containerized architecture for edge computing in smart home: A consistent architecture for model deployment," in *2020 International Conference on Computer Communication and Informatics (ICCCI)*, pp. 1–8, IEEE, 2020.

[171] P. Pääkkönen and D. Pakkala, "Extending reference architecture of big data systems towards machine learning in edge computing environments," *Journal of Big Data*, vol. 7, no. 1, pp. 1–29, 2020.

[172] "Next Generation Hybrid Cloud Data Analytics Solutionb." `https://builders.intel.com/docs/datacenterbuilders/next-gen-hybrid-cloud-data-analytics-solution.pdf`.

[173] D. Brayford, S. Vallecorsa, A. Atanasov, F. Baruffa, and W. Riviera, "Deploying ai frameworks on secure hpc systems with containers.," in *2019 ieee high performance extreme computing conference (hpec)*, pp. 1–6, IEEE, 2019.

[174] "Machine Learning Lens-AWS Well-Architected Frameworkb." `https://d1.awsstatic.com/whitepapers/architecture/wellarchitected-Machine-Learning-Lens.pdf`.

[175] S. Tuli, N. Basumatary, and R. Buyya, "Edgelens: Deep learning based object detection in integrated iot, fog and cloud computing environments," in *2019 4th International Conference on Information Systems and Computer Networks (ISCON)*, pp. 496–502, IEEE, 2019.

[176] X. Zhang, Y. Wang, S. Lu, L. Liu, W. Shi, *et al.*, "Openei: An open framework for edge intelligence," in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 1840–1851, IEEE, 2019.

[177] J. Lin and A. Kolcz, "Large-scale machine learning at twitter," in *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pp. 793–804, 2012.

[178] 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*, pp. 1387–1395, 2017.

[179] D. Crankshaw, P. Bailis, J. E. Gonzalez, H. Li, Z. Zhang, M. J. Franklin, A. Ghodsi, and M. I. Jordan, "The missing piece in complex analytics: Low latency, scalable model management and serving with velox," *arXiv preprint arXiv:1409.3809*, 2014.

[180] C. Olston, N. Fiedel, K. Gorovoy, J. Harmsen, L. Lao, F. Li, V. Rajashekhar, S. Ramesh, and J. Soyke, "Tensorflow-serving: Flexible, high-performance ml serving," *arXiv preprint arXiv:1712.06139*, 2017.

[181] A. Miklosik, M. Kuchta, N. Evans, and S. Zak, "Towards the adoption of machine learning-based analytical tools in digital marketing," *IEEE Access*, vol. 7, pp. 85705–85718, 2019.

[182] I. Karamitsos, S. Albarhami, and C. Apostolopoulos, "Applying devops practices of continuous automation for machine learning," *Information*, vol. 11, no. 7, p. 363, 2020.

[183] B. S. Farroha and D. L. Farroha, "A framework for managing mission needs, compliance, and trust in the devops environment," in *2014 IEEE Military Communications Conference*, pp. 288–293, IEEE, 2014.

[184] B. Fitzgerald and K.-J. Stol, "Continuous software engineering: A roadmap and agenda," *Journal of Systems and Software*, vol. 123, pp. 176–189, 2017.

[185] J. Bosch, "Continuous software engineering: An introduction," in *Continuous software engineering*, pp. 3–13, Springer, 2014.

[186] M. Leppänen, S. Mäkinen, M. Pagels, V.-P. Eloranta, J. Itkonen, M. V. Mäntylä, and T. Männistö, "The highways and country roads to continuous deployment," *Ieee software*, vol. 32, no. 2, pp. 64–72, 2015.

[187] I. Weber, S. Nepal, and L. Zhu, "Developing dependable and secure cloud applications," *IEEE Internet Computing*, vol. 20, no. 3, pp. 74–79, 2016.

[188] "Continuous Delivery vs. Continuous Deploymentb." `https://continuousdelivery.com/2010/08/continuous-delivery-vs-continuous-deployment`. Accessed: 2021-03-25.

[189] L. Chen, "Continuous delivery: Huge benefits, but challenges too," *IEEE software*, vol. 32, no. 2, pp. 50–54, 2015.

[190] "What is Continuous Delivery?b." https://continuousdelivery.com/2010/02/continuous-delivery/. Accessed: 2021-03-16.

[191] "MLOps: Continuous delivery and automation pipelines in machine learning." https://cloud.google.com/solutions/machine-learning/mlops-continuousdelivery-and-automation-pipelines-in-machine-learning. Accessed: 2021-03-16.

[192] R. Yin, "Case study research and applications: design and methods: sage publications," 2018.

[193] Y. Zhou, Y. Yu, and B. Ding, "Towards mlops: A case study of ml pipeline platform," in *2020 International Conference on Artificial Intelligence and Computer Engineering (ICAICE)*, pp. 494–500, IEEE, 2020.

[194] J. Lim, H. Lee, Y. Won, and H. Yeon, "Mlop lifecycle scheme for vision-based inspection process in manufacturing," in *2019 {USENIX} Conference on Operational Machine Learning (OpML 19)*, pp. 9–11, 2019.

[195] "MLOps: Continuous Delivery for Machine Learning on AWS.b." https://d1.awsstatic.com/whitepapers/mlops-continuous-delivery-machine-learning-on-aws.pdf. Accessed: 2021-03-25.

[196] "Onicab." https://insights.onica.com/hubfs/2020%20Offerings/2020\protect\@normalcr\relax_Onica_MLOps_Foundations_Offer.pdf. Accessed: 2021-03-25.

[197] J. Gerring, "What is a case study and what is it good for?," *American political science review*, vol. 98, no. 2, pp. 341–354, 2004.

[198] "Machine Learning Operations (MLOps) A Meetup.b." https://higherlogicdownload.s3.amazonaws.com/IMWUC/UploadedImages\protect\@normalcr\relax/12f13a33-bced-4573-8dd7-be58d519757c/MLOps.pdf. Accessed: 2021-03-25.

[199] "MLOpsb." https://perspecta.com/sites/default/files/2021-02/MLOps%20Whitepaper_022321.pdf. Accessed: 2021-03-25.

[200] "MLOps with Azure AutoML.b." https://query.prod.cms.rt.microsoft.com\protect\@normalcr\relax/cms/api/am/binary/RE4KW7y. Accessed: 2021-03-25.

[201] A. Banerjee, C.-C. Chen, C.-C. Hung, X. Huang, Y. Wang, and R. Chevesaran, "Challenges and experiences with mlops for performance diagnostics in hybrid-cloud enterprise software deployments," in *2020 {USENIX} Conference on Operational Machine Learning (OpML 20)*, 2020.

[202] "MLOps stackb." https://valohai.com/assets/files/mlops-stack.pdf. Accessed: 2021-03-25.

[203] "MLOps using MLFlowb." https://www.iteblog.com/ppt/dataai-summit-europe-2020/mlops-using-mlflow-iteblog.com.pdf. Accessed: 2021-03-25.

[204] "MLOps Foundationsb." https://www.rackspace.com/sites/default/files/white-papers/MLOps_Foundations_Data_Sheet.pdf. Accessed: 2021-03-25.

[205] "MLOps: Machine Learning Operationalization.b." https://cdn.activestate.com/wp-content/uploads/2018/10/webinar-slides-mlops.pdf. Accessed: 2021-03-25.

[206] "What is MLOps. b." https://blogs.bmc.com/mlops-machine-learning-ops/?print=pdf. Accessed: 2021-03-25.

[207] "Eight must-haves for MLOps success and when to use them.b." https://info.algorithmia.com/hubfs/2020/Webinars/Forrester/Eight-must-haves-slides-final.pdf?hsLang=en-u. Accessed: 2021-03-25.

[208] "Deoliteb." https://www2.deloitte.com/content/dam/Deloitte/us/Documents\protect\@normalcr\relax/technology/us-ml-oops-to-mlops.pdf. Accessed: 2021-03-25.

[209] "MLOps: Industrialized AI.b." https://www2.deloitte.com/content/dam/insights\protect\@normalcr\relax/

articles/7022_TT-MLOps-industrialized-AI/DI_2021-TT-MLOps_infographic.pdf. Accessed: 2021-03-25.

[210] L. C. Silva, F. R. Zagatti, B. S. Sette, L. N. dos Santos Silva, D. Lucrédio, D. F. Silva, and H. de Medeiros Caseli, "Benchmarking machine learning solutions in production," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 626–633, IEEE, 2020.