Machine Learning Behind The Scenes: An Exploratory Study in Fintech

Mark Haakman†, Luís Cruz‡, Hennie Huijgens‡, Arie van Deursen‡
Software Engineering Research Group, Delft University of Technology
Delft, The Netherlands
Email: †m.p.a.haakman@student.tudelft.nl, ‡{l.cruz, h.k.m.huijgens, arie.vandeursen}@tudelft.nl

***Abstract*—Artificial Intelligence has become increasingly important for organizations. Pioneers in the Artificial Intelligence industry are asking how to better develop and maintain Artificial Intelligence software. This paper focuses on Machine Learning, the branch of Artificial Intelligence that deals with the automatic generation of knowledge models based on sample data. This study aims to understand the evolution of the processes by which Machine Learning applications are developed and how state-of-the-art lifecycle models fit the current needs of the fintech industry. We conducted an exploratory case study at ING, a global bank with a strong European base. We interviewed 17 people with different roles and from different departments within the organization. We analyze existing lifecycle models in the literature and refine them by adding stages for data collection, a feasibility study, documentation, model monitoring, and a sub-stage of model risk assessment within model evaluation. The results indicate that the existing lifecycle models CRISP-DM and TDSP largely reflect the current development processes of Machine Learning applications, but there are crucial steps missing from the fintech perspective, including a feasibility study, documentation, model evaluation, and model monitoring. Our work shows that the real challenges of applying Machine Learning go much beyond sophisticated learning algorithms – more focus is needed on the entire lifecycle.**

***Index Terms*—Machine Learning Lifecycle, Case Study, Fintech**

I. Introduction

Artificial Intelligence (AI) has become increasingly important for organizations to support customer value creation, productivity improvement, and insight discovery. Pioneers in the AI industry are asking how to better develop and maintain AI software [1]. This paper focuses on Machine Learning, the branch of AI that deals with the automatic generation of knowledge models based on sample data.

Although most of the AI techniques are not so recent (e.g., neural networks were already being applied in the 1980s [2]), the recent access to large amounts of data and more computing power has exploded the number of scenarios where AI can be applied [3], [4]. In fact, AI is now being used to add value in critical business scenarios. Consequently, a number of new challenges are emerging in the lifecycle of AI systems, comprising all the stages from their conception to their retirement and disposal. Like normal software applications, these projects need to be planned, tested, debugged, deployed, maintained, and integrated into complex systems.

Companies leading the advent of AI are reinventing their development processes and coming up with new solutions. Thus, there are many lessons to be learned to help other organizations and guide research in a direction that is meaningful to the industry. This is particularly relevant for highly-regulated industries such as fintech, as new processes need to be designed to make sure AI systems meet all required standards.

Recent research has addressed how developing AI systems is different from developing regular Software Engineering systems. A case study at Microsoft identified the following differences [5]: 1) data discovery, management, and versioning are more complex; 2) practitioners ought to have a broader set of skills; and 3) modular design is not trivial since AI components can be entangled in complex ways. Unfortunately, existing research offers little insight into the challenges of transforming an existing IT organization into an AI-intensive one.

Examples of existing models describing the Machine Learning lifecycle are the Cross-Industry Standard Process for Data Mining (CRISP-DM) [6] and the Team Data Science Process (TDSP) [7]. However, Machine Learning is being used for different problems across many different domains. Given the fast pace of change in AI and recent advancements in Software Engineering, we suspect that there are deficiencies in these lifecycle models when applied to a fintech context.

To remedy this, we set out this exploratory case study aimed at understanding and improving how the fintech industry is currently dealing with the challenges of developing Machine Learning applications at scale. ING is a relevant case to study, since it has a strong focus on financial technology and Software Engineering and it is undergoing a bold digital transformation to embrace AI as an important competitive factor. By studying ING, we provide a snapshot of the rapid evolution of the approach to Machine Learning development.

We define the following research questions for our study:

***RQ1**: How do existing Machine Learning lifecycle models fit the fintech domain?*

***RQ2**: What are the specific challenges of developing Machine Learning applications in fintech organizations?*

We interviewed 17 people at ING with different roles and from different departments. Thereafter, we triangulated the resulting data with other resources available inside the organization. Furthermore, we refine the existing lifecycle models CRISP-DM and TDSP based on our observations at ING.

Our results unveil important challenges that ought to be addressed when implementing Machine Learning at scale. Feasibility assessments, documentation, model risk assessment, and model monitoring are stages that have been overlooked by existing lifecycle models. There is a lack of standards and there is a need for automation in the documentation and governance of Machine Learning models. Finally, we pave the way for shaping the education of AI to address the current needs of the industry.

The remainder of this paper is structured as follows. In Section II we introduce existing lifecycle models and describe related work. In Section III, we outline the study design. We report the data collected in Section IV and we answer the research questions in Section V. We discuss our findings and threats to validity in Section VI. Finally, in section VII, we pinpoint conclusions and outline future work.

II. Background

In this section, we present the lifecycle models considered in this study and examine existing literature outlining the differences with our study.

A. Existing Lifecycle Models

In this study, we consider two reference models for the lifecycle of Machine Learning applications: Cross-Industry Standard Process for Data Mining (CRISP-DM) [6] and Team Data Science Process (TDSP) [7]. We chose CRISP-DM, as although it is twenty years old, it is still the *de facto* standard for developing data mining and knowledge discovery projects [8]. We selected TDSP as modern industry methodology, which has at a high level much in common with CRISP-DM. There are other methodologies, but most are similar to CRISP-DM and TDSP. Findings in our paper can be extrapolated to those other methodologies.

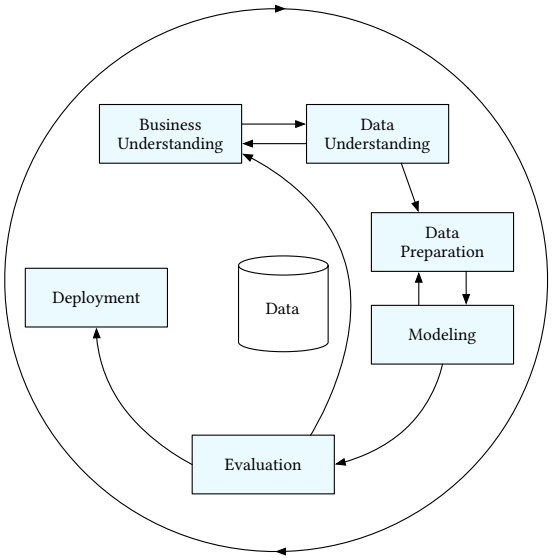

Fig. 1. Cross-Industry Standard Process for Data Mining (CRISP-DM).

CRISP-DM aims to provide anyone with "a complete blueprint for conducting a data mining project" [6]. Although data mining is not the common term used nowadays, it is valid for any project applying scientific methods to extracting value from data, including Machine Learning [8]. CRISP-DM breaks down a project into six phases, as presented in Figure 1. It typically starts with **Business Understanding** to determine business objectives, going back and forward with **Data Understanding**. It is followed by **Data Preparation** to make data ready for **Modeling**. The produced model goes through an **Evaluation** in which it is decided whether the model can go for **Deployment** or it needs another round of improvement. The arrows between stages indicate the most relevant and recurrent dependencies, while the arrows in the outer circle indicate the evolution of Machine Learning systems after being deployed and their iterative nature.

Based on CRISP-DM, a number of lifecycle models have been proposed [8], [9] to address varying objectives. Derived models extend CRISP-DM for projects with geographically dispersed teams [10], with large amounts of data and more focus on automation [11], [12], or targeting the model reuse across different contexts [13].

TDSP is "an agile, iterative data science methodology" by Microsoft, to deliver Machine Learning solutions efficiently [7]. The original methodology includes four major stages, as can be seen in Figure 2: **Business Understanding**, **Data Acquisition**, **Modeling** and **Deployment**. As depicted by the arrows in the figure, TDSP proposes stronger dependencies but does not enforce a particular order between stages, emphasizing that different stages can be iteratively repeated at almost any time in the project.

Despite the number of advancements proposed in previous work, they do not tackle AI systems that target challenges faced by the fintech industry. Our work pinpoints the changes that needed to be accommodated for AI systems operating under heavy-regulated scenarios and bringing value over pre-existing non-data-driven approaches.

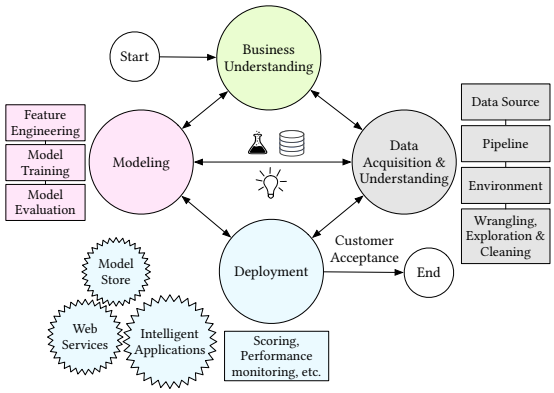

Fig. 2. Team Data Science Process (TDSP).

B. Related Work

The Machine Learning development lifecycle has been studied in practice in previous research. Amershi et al. [5] have conducted a case study at Microsoft to study the differences between Software Engineering and Machine Learning. The most important challenges found are model scaling, evolution, evaluation, deployment, and data management. We complement this study by comparing our observations with existing Machine Learning lifecycle models.

Another case study from industry has been performed at Booking.com by Bernardi et al. [4]. In contrast with academic research in which Machine Learning models are validated by means of an error measurement, models at Booking.com are validated through business metrics such as conversion or cancellations. The paper describes process stages such as model designing, deployment, monitoring, and evaluation, but no formal lifecycle model is defined.

Hill et al. [14] studied how people develop intelligent systems in practice. The study leverages a high-level model of the process and identifies the main challenges. Results show that developers struggle with establishing repeatable processes and that there is a basic mismatch between the tools available versus the practical needs. In this study, we extend the work by Hill et al. by looking more closely at what happens after the Machine Learning model has been evaluated, for example regarding its deployment and monitoring.

The paper by Lin and Ryaboy [15] describes the *big data mining cycle* at Twitter, based on the experience of the two authors. The main points made are that for data-driven projects, most time goes to preparatory work before, and engineering work after the actual model training and that a significant amount of tooling and infrastructure is required. In our study, we validate the recommendations of these two experts with a case study with seventeen participants.

Concrete challenges data scientists face are elaborated upon in the study by Kim et al. [16]. They have surveyed 793 professional data scientists at Microsoft. An example of a challenge found is that the proliferation of data science tools makes it harder to reuse work across teams. This challenge is also reinforced in the study by Ahmed et al. [17]. As models are mostly implemented without standard API, input format, or hyperparameter notation, data scientists spend considerable effort on implementing glue code and wrappers around different algorithms and data formats to employ them in their pipelines. Ahmed et al. [17] show evidence that most models need to be rewritten by a different engineering team for deployment. The root of this challenge lies on runtime constraints, such as a different hardware or software platform, and constraints on the pipeline size or prediction latency.

More studies looked at Machine Learning from a Software Engineering viewpoint. Sculley et al. [18] identified a number of Machine Learning-specific factors that increase technical debt, such as boundary erosion and hidden feedback loops. Breck. et al [19] have proposed 28 specific tests for assessing production readiness for Machine Learning applications. These tests include tests for features and data, model development, infrastructure, and monitoring. Arpteg et al. [20] have identified Software Engineering challenges of building intelligent systems with deep learning components based on seven projects from companies of different types and sizes. These challenges include development, production, and organizational challenges, such as experiment management, dependency management, and effort estimation. In this current study, we will extend this line of research and identify where Software Engineering can help mitigate inefficiencies in the development and evolution of Machine Learning systems.

III. Research Design

To identify the gaps in the existing Machine Learning lifecycle models and explore key challenges in the field, we perform a single-case exploratory case study. This is a recurrent methodology to define new research by looking at concrete situations and to shed empirical light on existing concepts and principles [21]. We follow the guidelines proposed by Brereton et al. [22] and Yin's [21] case study methodology.

It is not our objective to build an entirely new theory from the ground up. For that reason, we do not adopt a Grounded Theory (GT) approach, although we do use a number of techniques based on GT [23]: e.g., theoretical sampling, memoing, memo sorting, and saturation.

The design of the study is further described in this section.

A. The Case

The case under study is ING, a global bank with a strong European base. ING offers retail and wholesale banking services to 38 million customers in over 40 countries, with over 53,000 employees [24]. ING has a strong focus on fintech, the digital transformation of the financial sector, and professionalization of AI development.

A bank of this size has many use cases where Machine Learning can help. Examples include traditional banking activities such as assessing credit risk, the execution of customer due diligence and transaction monitoring requirements related to fighting financial economic crime. Other examples of use cases are improving customer service and IT infrastructure monitoring.

ING is currently leveraging a major shift in the organization to adopt AI to improve its services and increase business value. The challenges that ING is facing at the moment make it an interesting case for our study and allow us to identify gaps between current challenges by the industry and academia.

B. Research Methodology

Semi-structured interviews are the main source of data in this case study. The data is later triangulated with other resources available inside the organization. Documentation in the intranet of ING is used to gain a deeper understanding of the platforms and processes mentioned in the interviews.

The approach used to collect information from interviews and report data is based on the guidelines proposed by Halcomb et al. [25]. It is a reflexive, iterative process:

TABLE I
OVERVIEW OF INTERVIEWEES

ID	Role	Department
P01	IT Engineer	Application Platforms
P02	IT Engineer	IT Infrastructure Monitoring
P03	Productmanager	Financial Crime
P04	IT Architect	Enterprise Architects
P05	IT Engineer	IT4IT
P06a*	Advice Professional	Model Risk Management
P06b*	Advice Professional	Model Risk Management
P07	Manager IT	Global Engineering Platform
P08	Feature Engineer	Data & Analytics
P09	Data Scientist	Wholesale Banking Analytics
P10	Data Scientist	Chapter Data Scientists
P11	IT Engineer	Application Platform
P12	Data Scientist	AIOps
P13	Data Scientist	Wholesale Banking Analytics
P14	Data Scientist	Financial Crime
P15	Data Scientist	Analytics
P16	Data Scientist	Chapter Data Scientists

**The sixth interview involved two participants, labeled P06a and P06b.*

1) Audio taping of the interview and concurrent note-taking.
2) Reflective journaling immediately post-interview.
3) Listening to the audiotape and revising memos.
4) Data analysis.

*1) **Participants**:* We selected interviewees based on their role and their involvement in the process of developing Machine Learning applications. We strove to include people of many different roles and from many different departments. The starting position for finding interviewees was the lead of a Software Analytics research team within ING. More interviewees were found by the recommendations of other interviewees. The interviewees were also able to suggest other sources of evidence that might be relevant. We increased the number of participants until we reached a level of saturation in the remarks mentioned by interviewees for each stage of the lifecycle.

An overview of the selected participants, with their role and department, can be seen in Table I. In total, we interviewed seventeen participants. The sixth interview involved two participants. Therefore, they are labeled as P06a and P06b.

*2) **Interview Design**:* The first two authors conducted the interviews, which took approximately one hour. We took notes during the interviews and we recorded the interviews with the permission of the participants. This section outlines the main steps of our interview design. The full details can be found in our corresponding case study protocol [26].

As interviewers, we started by introducing ourselves and provided a brief description of the purpose of the interview and how it relates to the research being undertaken. We asked the interviewees to introduce themselves and describe their main role within the organization. After the introductions, we asked the interviewee to think about a specific Machine Learning project he or she was working on recently. Based on that project, we asked the interviewee to describe all the different stages of the project. In particular, we asked questions to understand the main challenges they faced and the solutions they had to design.

*3) **Post-interview Strategy**:* Right after each interview, the two interviewers got together for a collaborative *memoing* process (also called *reflective journaling* [25]). Memoing is the review and formalization of field-notes and expansion of initial impressions of the interaction with more considered comments and perceptions. Memoing is chosen over creating verbatim transcriptions, because the costs associated with interview transcription, in terms of time, physical, and human resources, are significant. Also, the process of memoing assisted the researchers to capture their thoughts and interpretations of the interview data [27]. The audio recordings could still be used to facilitate a review of the interviewers' performance, and assist interviewers to fill in blank spaces in their field notes and check the relationship between the notes and the actual responses [28].

The interviewers took between 30–45 minutes to refine their notes. In this process, the notes are assigned under different lifecycle stages. We used the nomenclature from existing frameworks (e.g. CRISP-DM and TDSP) as a rule of thumb, or we defined new stages in case it helps understand a particular part of the process.

After some time, the interviewers amended the memos by reviewing the audiotapes. The purpose of this stage was to ensure that the memos provided an accurate reflection of the interviews [25]. Once the researchers were confident that their memos accurately represented each interview, the process of content analysis is used to elicit common themes [25].

Each interview resulted in three artifacts: the recording of the interview, the field notes taken during the interview, and the memos as a result of the above mentioned memoing.

IV. DATA ANALYSIS

The input of the interviewees does not answer the research questions directly. Therefore, we report the resulting data of the interviews in this section and we use this data to answer the research questions later in Section V.

We organize the data among eight core Machine Learning lifecycle stages: problem design, requirements, data engineering, modeling, documentation, model evaluation, model deployment, and model monitoring. Overarching data that does not fit these stages is categorized under testing, iterative development, and education. These stages and categories are based on stages defined by CRISP-DM and TDSP (cf. Section II-B) or mentioned by practitioners themselves.

For all the remarks, we identify the practitioner who mentioned them by referencing the corresponding ID from Table I. Given that this is a qualitative analysis, the number of individuals supporting a particular result has no quantitative meaning on its relevance.

A. Problem Design

Machine Learning projects at ING start with the definition of the problem that needs to be solved. Two main approaches are observed in this study:

1) Innovation push: a stakeholder comes up with a question or problem that needs to be solved. A team is set up to design a solution using a suitable Machine Learning technique.
2) Technology push: a team identifies new data or a set of Machine Learning techniques that may add business value and are potentially useful or solving problems within the organization. This approach aims to optimize processes, reduce manual work, increase model performance, and create new business opportunities.

The problem is defined together with stakeholders and it is assessed whether using Machine Learning is appropriate to solve the problem (P01, P14, P15). In the teams of P15 and P14, this is done by collaboratively filling in a project document with the stakeholders which contains information like the problem statement, goals, and the corresponding business case. Also, domain experts outside the teams are part of this.

B. Requirements

Besides project-specific requirements, many of the requirements come from the organization and are applicable to every Machine Learning application (P15). These requirements include traceability, interpretability, and explainability (P01, P04, P07, P15). Together with all other regulatory requirements, they pose a big challenge while developing Machine Learning applications (P04). A natural consequence of regulatory requirements is that black-box AI models cannot be used in most situations (P01, P04, P14). For risk management safeness, only interpretable/explainable AI models are accepted.

Project-specific requirements are often defined by the product owner together with the stakeholders (P10). Data requirements are said to become more clear while working with the model (P04). As the users of the system are often no Machine Learning experts, defining the model performance requirements is sometimes a challenge (P09, P13).

C. Data Engineering

Interviewees describe that data engineering requires the major part of the lifetime of a Machine Learning project (P03, P10, P15) and is also the most important for the success of the project (P10).

1) ***Data Collection****:* Data collection is considered a very challenging and time-consuming task (P03, P04, P12, P14). Typical use cases require access to sensitive data, which needs to be formally requested. ING has an extensive data governance framework that, among others, assigns data management roles (e.g. data owner) and rules for obtaining, sharing, and using data. Each dataset is assigned a criticality rating, to define the degree of data governance and control required.

There might be people with different access privileges to data in the same project. This means that, in the exploratory stages of some projects using critical data, only a restricted number of team members (e.g., data scientists) are able to perform an exploratory analysis of data. The remaining practitioners will only have access to the model specification (P04).

A challenge of data collection is making sure that the (training and test) data collected is representative of the problem (P13). As an example, if a Machine Learning model is trained on systems logs, it should be made sure that logs of all systems are available. Another challenge is merging data from multiple sources (P10, P12). Going back to the logging example, different systems may have different logging formats, but the configurations of these formats can not be altered by the developers creating the model.

2) ***Data Understanding****:* In the data understanding stage, an assessment is done on the quality of available data and how much processing will be required to use that data. It comprises exploratory data analysis, often including graphical visualizations and summarization of data. According to P09, the temptation of applying groundbreaking Machine Learning techniques tends to overlook the importance of understanding the data.

Data understanding is also an important step to assess the feasibility of the project. Thus, it entails not only performing an exploratory analysis, but also a considerable effort in communicating the main findings to all the different stakeholders.

3) ***Data Preparation****:* After the data is collected and it is assessed that the data is representative of the problem being solved, the data is prepared to be used for modeling.

A challenge regarding data preparation is that the same preprocessing has to be ensured in the development environment and in the production environment (P08, P09). Data streams in production are different than in the development environment and it is easier to clean training and testing data than production data (P09).

D. Modeling

Model training is mostly done in on-premises environments such as Hadoop[1] and Spark[2] clusters (P09) or in generic systems using, for example, the scikit-learn[3] library (P01). These private platforms are connected with the data lakes where data is stored, so training can be done on (a copy of) real production data (P01, P03). The on-premises environment has no outgoing connection to the internet, so a connection to other cloud services such as Microsoft Azure[4] or Google Cloud[5] is not possible (P08). This means that data scientists are limited to the tools and platforms available within the organization when dealing with sensitive data. Also, all project dependencies need to be previously approved, after which they are made available in a private package repository (P12), which contains whitelisted packages that have been internally audited. Fewer restrictions are in place if Machine Learning

[1]Hadoop enables distributed processing of large data sets across clusters of computers Website: https://hadoop.apache.org.

[2]Spark is a unified analytics engine for large-scale data processing. Website: https://spark.apache.org.

[3]Scikit-learn is a Machine Learning library for Python. Website: https://scikit-learn.org.

[4]Microsoft Azure is a cloud computing service. Website: https://azure.microsoft.com/en-us.

[5]Google Cloud is a cloud computing service. Website: https://cloud.google.com.

is applied to public data, for example on stock prices. In that case, external cloud services and packages may be used (P09).

Model training is an iterative process. Usually, multiple models are created for the same problem. First, a simple model is created (e.g., a linear regression model) to set as a baseline (P09). In the following iterations, more advanced models are compared to this baseline model. If an approach other than Machine Learning already exists (e.g., rule-based software), the models are also compared with this.

To keep track of different versions of models, different teams use different strategies. For example, the team of P08 keeps track of an experiment log using a spreadsheet, in which the training set, validation set, model, and pre-processing steps are specified for each version. This approach for versioning is preferred over solutions like MLFlow[6] for the sake of simplicity (P08, P15).

*1) **Model Scoring**:* An implicit sub stage of modeling is assessing model performance to measure how well the predictions of the model represent ground truth data.

We define *Model Scoring* as assessing the performance of the model based on scoring metrics (e.g., f1-score for supervised learning). It is also known as *Validation* by the Machine Learning community, which should not be confused with the definition by the Software Engineering community[7] [29], [30].

The main remarks for this stage are related to defining the right set of metrics (P03, P06, P12, P14, P15, P16). The problem is two-fold: 1) identify the right metrics and 2) communicate why the selected metrics are right. Practitioners report that this is very problem-specific. Thus, it requires a good understanding of the business, data, and learning algorithms being used. From an organization's point of view, these different perspectives are a big barrier to defining validation standards.

E. Documentation

Each model has to be documented (P02). This serves multiple goals. It makes assessing the model from a regulatory perspective possible (P09, P13), it enables reproducibility, and also can make the model better because it is looked at from a broad perspective – i.e., a "helicopter view" (P09). It also provides an audit trail of actions, decisions, versions, etc. that supports evidencing. Documentation also supports the transfer of knowledge, for example, new team members or the end-users which are mostly not Machine Learning experts (P12). Just like code, documentation is also peer-reviewed (P13).

The content of the documentation differs slightly per department, but all documentation should at least follow the minimum standards defined by the model risk management framework (P06). Some teams extend on this by creating templates for documentation themselves (P13). In general, the following is documented when developing a Machine Learning application: the purpose, methodology, assumptions, limitations, and the use of the model. More concretely, a Technical Model Document is created which includes the model methodology, input, output, performance metrics and measurements, and testing strategy (P14). It furthermore states all faced difficulties and their solutions, plus the main (technical) decisions (P09). It has to explain why a certain model is chosen and what its inner workings are, to be able to demonstrate the application does what the creators claim it is doing.

F. Model Evaluation

An essential step in the evaluation of the model is communicating how well the model performs according to the defined metrics. It is about demonstrating that the model meets business and regulatory needs and assessing the design of the model. One key difference between the metrics used in this step and the metrics used for *Model Scoring* is that these metrics are communicated to different stakeholders that do not necessarily have a Machine Learning or data science background. Thus, the set of metrics needs to be extended to a general audience. One complementary strategy used by practitioners is having live demos of the model with business stakeholders (P03, P15, P16). These demos allow stakeholders to try out different inputs and try corner cases.

*1) **Model Risk Assessment**:* An important aspect of evaluating a model at ING is making sure it complies with regulations, ethics, and organizational values (P15, P06). This is a common task for any type of model built within the organization – i.e., not only Machine Learning models but also economic models, statistical forecasting models, and so on. In the interviews, *Model Risk Assessment* was mentioned as mandatory within the model governance strategy, undertaken in collaboration with an independent specialized team (P06, P14). Depending on the criticality level of the model, the intensity of the review may vary. Each model owner is responsible for the risk management of their model, but colleagues from the risk department help and challenge the model owner in this process.

During the periodic risk assessment process, assessors inspect the documentation provided by the Machine Learning team to assess whether all regulations and minimum standards are followed. Although the process is still under development within ING, the following key points are being covered: 1) model identification (identify if the candidate is a model which needs risk management), 2) model boundaries (define which components are part of the model), 3) model categorization (categorize the model into the group of models with a comparable nature, e.g. anti-money-laundering), 4) model classification (classify the model into in the class of models which require a comparable level of model risk management), and 5) assess the model by a number of sources of risk.

G. Model Deployment

We observed three deployment patterns at ING:

[6]MLFlow is a platform to manage the Machine Learning lifecycle. Website: https://mlflow.org.

[7]*Validation* in Software Engineering "is the set of activities ensuring and gaining confidence that a system is able to accomplish its intended use, goals and objectives" [29].

1) A specialized team creates a prototype with a validated methodology, and an engineering team takes care of reimplementing it in a scalable, ready-to-deploy fashion. In some cases, this is a necessity due to the technical requirements of the model, e.g., when models are developed in Python, but should be deployed in Java (P08, P09, P13).
2) A specialized team creates a model and exports its configuration (e.g., a *pickle*[8] and required dependencies) to a system that will semi-automatically bundle it and deploy it without changing the model (P01, P09).
3) The same team takes care of creating the model and taking it into production. This mostly means that software engineers are part of the team and a structured and strict software architecture is ensured.

Similar to the training environments, Machine Learning systems are deployed to on-premises environments. A reported challenge regarding the deployment environment is that different hardware and platform parameters (e.g., Spark parameters) can result in different model behavior or errors (P16). For example, the deployment environment may have less memory than the training environment. Furthermore, the resources for a Machine Learning system are dynamically allocated whenever needed. However, it is not trivial understanding when a system is no longer needed and should be scaled down to zero (P01).

H. Model Monitoring

After having a model in production, it is necessary to keep track of its behavior to make sure it operates as expected. It implies testing the model while the model is deployed online. The main advantage is that it uses real data. Previous work refers to this stage as *online testing* [**?**].

The inputs and outputs of the model are monitored while it is executing. Each model requires a different approach and different metrics, as standards are not yet defined. In this stage, practitioners also look into whether the statistical properties of the target variable do not change in unforeseen ways (P11). The model behavior is mostly monitored by data science teams and is still lacking automation (P03, P05, P06, P14). Also the impact on user experience is monitored when the model has a direct impact on users. This is mostly done using A/B testing techniques and can have business stakeholders directly involved (P03, P10).

Teams resort to self-developed or highly-customized dashboard platforms to monitor the models (P15, P16). Within the organization, different teams may have different platforms. While standardization is in development, for now, we have not observed solutions that are used across the organization. A big challenge in making these platforms available is the fact that each problem has different monitoring requirements and considerable engineering efforts need to be undertaken to effectively monitor a given model and implement access privileges (P15).

[8]A *pickle* is a serialized Python object. Website: https://docs.python.org/3/library/pickle.html.

I. Testing

Testing is a task that is transversal to the whole development process. It is done at the model level and at the software level.

Testing at the model level addresses requirements such as correctness, security, fairness, and interpretability. With the exception of correctness, we have not observed automated approaches to verify these requirements. A challenge for the correctness tests is defining the number of errors that are acceptable – i.e., the right threshold (P14).

For testing at the software level, unit and integration testing is the general approach. It scopes any software used in the lifecycle of the model (P07). It enables the verification of whether the techniques adopted in the design of the Machine Learning system are working as expected. However, although unit and integration testing is part of the checklist used for *Model Evaluation*, a number of projects are yet not doing it systematically (P12, P15). As reported by P14, tests are not always part of the skill set of a data scientist. Nevertheless, there is a generalized interest in learning code testing best practices (P12).

J. Iterative Development

At ING, teams adopt agile methodologies. Three practitioners (P03, P09, P14) mentioned that using agile methodologies is not straightforward in the early phases of Machine Learning projects. They argued that performing a feasibility study does not fit in small iterations. The first sprint requires spending a considerable amount of time understanding and preparing data before being able to deliver any model. On the other hand, interviewees acknowledge the benefits of using agile (P03, P14). It helps keep the team focused on practical achievements and goals. Another advantage is that stakeholders are kept in the loop (P14).

Typically 2–3 data scientists are working together on the same model. For this reason, issues with having many developers working on the same model and merging different versions of a model have not been disruptive yet.

1) ***Feasibility Study****:* The end of the first iteration is also a decisive step in the project. Based on the outcome of this iteration there is a go/no-go assessment with all the stakeholders, in which the project is evaluated in terms of *viability* (i.e., does it solve a business issue), *desirability* (i.e., is it complying with ethics or governance rules), and *feasibility* (i.e., cost-effectiveness) (P04, P09, P15, P16). This process is well-defined within the organization for all innovation projects. According to P04 and P09, feasibility assessments are essential at any point of the project – it is important to adopt a *fail-fast* approach.

K. Education

Interviewees indicated multiple ways in which education can be improved to make graduates better Machine Learning practitioners in the industry. Firstly, data scientists should have more knowledge of Software Engineering and vice-versa (P01, P11, P14, P16). P11 indicates that data scientists with little software engineering knowledge will produce code that is

harder to maintain and likely increases technical debt. On the other hand, a software engineer without data science expertise may write clean code, which nevertheless may not add much business value, because of ineffective data exploration strategies (P09).

Another remark by practitioners is that education should put more focus on the process instead of techniques (P08). While graduates are appreciated for their broad sense of the state-of-the-art, they must learn how to tackle Machine Learning issues in large organizations (P08, P10). Academia knows well how to work with new projects, but in reality, the history of the company affects how to perform Machine Learning – e.g., integration with legacy systems (P08). Graduates seem to underestimate the efforts needed for data engineering, especially data collection (P03, P09, P12). Also, too much attention lies solely on the performance of models. In reality, over-complex models cannot be applied in organizations, because they tend to be too slow or too hard to explain (P16). These models – squeezing every bit of performance – are great for data science competitions as facilitated on Kaggle, but not for the industry, where more efficient solutions are necessary (P09, P16).

V. Data Synthesis

In this section, we answer each research question.

***RQ1**: How do existing Machine Learning lifecycle models fit the fintech domain?*

To answer this research question, we analyze lifecycle models existing in the literature and adapt them according to the findings observed in our study. We select two reference models, as described in Section II-A: *CRISP-DM* [6] and *TDSP* [7]. The changes we propose can be constrained to this specific case of ING, the fintech domain, or be extendable to general Machine Learning projects. We justify and define these constrains for each change.

Most stages we observed at ING naturally fit CRISP-DM and TDSP. Similarities between CRISP-DM and the lifecycle of Machine Learning models at ING are *Business Understanding*, *Data Understanding*, *Data Preparation*, *Modeling*, *Evaluation*, *Deployment*. Similarities between TDSP and the lifecycle at ING are *Business Understanding*, *Data Acquisition & Understanding*, *Modeling*, and *Deployment*. Nevertheless, based on the observations collected in our study, changes to these models are called for.

We propose the changes of CRISP-DM in Figure 3. We add three new essential stages: *Data Collection* (as part of *Data Engineering*), *Documentation*, and *Model Monitoring*. Furthermore, we emphasize the feasibility assessment with the "Go/No-go" checkpoint and a sub-stage *Model Risk Assessment*, part of *Evaluation*.

As depicted in Figure 4, we adapt the TDSP model to include *Documentation*, *Model Evaluation*, and *Model Monitoring* as major stages. We also emphasize *Model Risk Assessment* (as part of *Evaluation*) and a *Feasibility Study*.

The adaptations of the models will be further elaborated upon in the following paragraphs.

a) Data Collection: Although CRISP-DM encompasses *Data Collection* within *Data Understanding* and *Data Preparation*, our observations reveal important tasks and challenges that need to be highlighted. As reported in Section IV-C1, *Data Collection* requires getting privileges to access data with different criticality-levels and making sure the data is representative of the problem being tackled. Our proposition is that the characteristics observed at ING regarding this phase generalize to any large organization dealing with sensitive data.

b) Go/No-go or Feasibility Study: The aforementioned *Feasibility study* (cf. Section IV-J) is an essential part of a Machine Learning project to ensure projects have everything in place to deliver the long-term expectations. It was a recurrent step observed in our study, which is aligned with the agile approach, *Fail Fast*, promoted at ING and many organizations alike. It may generalize to other cases, depending on the agile culture of the organization.

c) Documentation: In our case, documentation revealed to be a quintessential artifact for a Machine Learning project. Documentation is the key source of knowledge on how the model is designed, evaluated, tested, deployed, and so on. The documentation is used to evaluate, maintain, debug, and keep track of any other decision regarding the model. It is hard to replace documentation with other strategies because stakeholders with a non-technical background also need to understand the model and have confidence in how the Machine Learning model is designed. Although documentation is also important in traditional Software Engineering applications, the codebase is usually the main target of analysis from audits. In Machine Learning, documentation contains important problem-specific decisions that cannot be understood in the code itself. We have no evidence on how this stage generalizes to other organizations, but believe this to be crucial in any highly regulated environment.

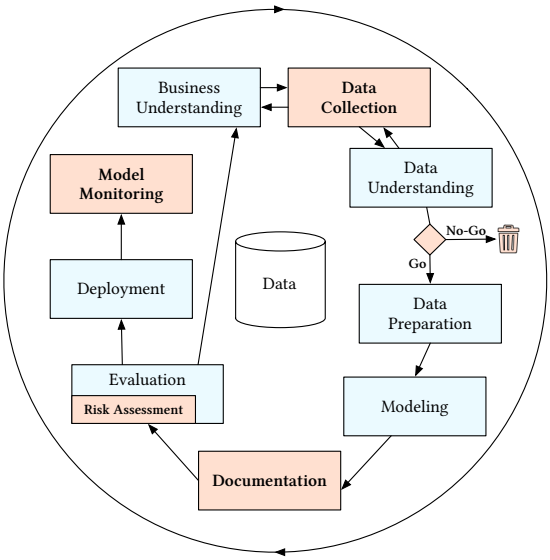

Fig. 3. Refined CRISP-DM model. Additions in red, with bold text.

d) Model Evaluation: Although the original version of TDSP also included *Model Evaluation*, it was proposed as an activity under the *Modeling* stage. We observed that, when we refer to assessing the performance of a model (i.e., *Model Scoring*), it is indeed part of the *Modeling* activities. However, there is an important part of the evaluation that requires more stable versions of the models. Moreover, it is undertaken with stakeholders that are not part of the *Modeling* loop – e.g., live demos with business managers (cf. Section IV-F). Thus, we highlight this part of the evaluation as its own stage. This is also relevant for projects in different domains.

e) Model Risk Assessment: Model Risk Assessment is crucial to any banking or finance organization. Although these companies already have a big history of traditional risk management, it does not cover Machine Learning models. At ING, this is mandatory for any model.

f) Model Monitoring: Most Machine Learning models operate continuously and produce outputs online. Our study shows that the natural step after deployment is *Monitoring* – for example, using dashboards – to ensure the model is behaving as expected. *Model Monitoring* is not explicit in neither CRISP-DM nor TDSP, but it is relevant to any domain.

Finally, although not depicted in the proposed lifecycles, *Education* is a stage implicit throughout the whole lifecycle. We observe that universities and courses on Machine Learning need to provide a more holistic approach to focus on all the different stages of the lifecycle of a Machine Learning system.

A lifecycle stage that we did not yet observe is the end of life of a Machine Learning system – i.e., the *Disposal* stage. We presume that a disposal stage is not relevant yet due to the recency of Machine Learning in fintech.

***RQ2**: What are the specific challenges of developing Machine Learning applications in fintech organizations?*

We highlighted many challenges of developing Machine Learning applications in Section IV. Most challenges fit in the CRISP-DM and TDSP models. However, two challenges specifically related to fintech and to our extensions of CRISP-DM and TDSP stand out: 1) *Model Governance* and 2) *Technology Access*.

Model Governance is on top of the agenda of the case in this study. A well-defined process is in place to validate regulations, ethics, and social responsibility in every Machine Learning model. The relevance of this problem to fintech organizations goes beyond Machine Learning applications: math-based financial models have long been deployed under well-defined risk management processes. Nevertheless, AI brings the need to revise and recreate model governance that suits the particularities of models that are now automatically trained. Model risk experts are now required to have a strong background in two disjoints fields: 1) *Governance, Risk Management, and Compliance* and 2) *AI*.

Technology Access is the second big challenge in developing AI in fintech organizations. All AI technologies, tools, and libraries need to be audited to make sure they are safe to be used in fintech applications. However, the field of AI is changing very fast with new tools. Industries that want to shift towards AI-based systems need to be able to quickly, yet safely, adopt new technologies.

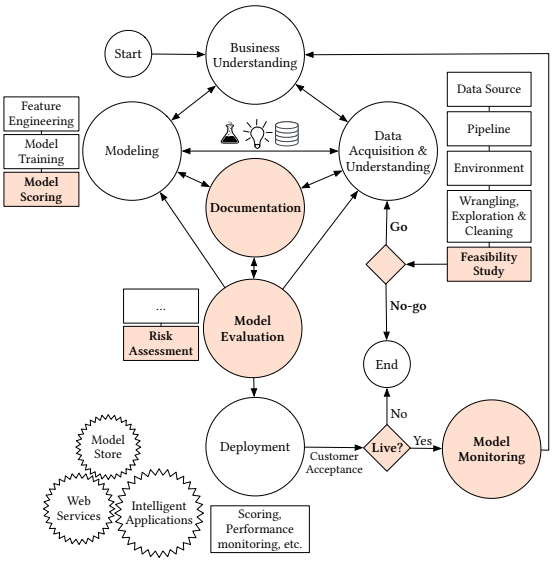

Fig. 4. Refined TDSP model. Additions in red, with bold text.

VI. Discussion

A. Implications

We see the following implications of our results for the fintech industry and for research.

*1) **Implications for Machine Learning Practitioners**:* Machine Learning practitioners have to be aware of extra steps and challenges in their process of developing Machine Learning applications. Although not mentioned in existing lifecycle models, the undertaking of feasibility assessments, documentation, and model monitoring, are crucial while developing Machine Learning applications.

*2) **Implications for Process Architects**:* Existing lifecycle models provide a canonical overview of the multiple stages in the lifecycle of a Machine Learning application. However, when being applied to a particular context, such as fintech, these models need to be adapted. From our findings, we suspect that this is also the case for other fields where AI is getting increasing importance. Process architects for intelligent systems for healthcare, autonomous driving, among many others, need to look at their lifecycle models from a critical perspective and update the models accordingly.

*3) **Implications for Researchers**:* Researchers could focus on solving the reported challenges in the Machine Learning lifecycle with additional tool support and reveal challenges of the ML lifecycle in other domains by extending the case study to more organizations and different types of industries.

More automation is required for exploratory data analysis and data integration techniques. Automation tools are also needed to help trace documentation back to the codebase and vice versa. Tools that assist model governance will reduce

bottlenecks in the development process and will help to ensure Machine Learning models comply with regulations.

Furthermore, solutions to challenges in the ML lifecycle should be researched. Software testing needs to be extended and adapted for Machine Learning software to help effectively test the Machine Learning pipeline at software-, data-, and model-level. It is also necessary to create holistic monitoring solutions that can scale to different models in an organization. There is a need for strategies to help practitioners select the right set of model scoring metrics. Agile development practices need to be adjusted for AI projects. Tools featuring experiment logs (e.g., MLFlow) ought to propose a holistic solution for version control to keep track of changes in data, changes in scoring metrics, and executions of different experiments.

4) ***Implications for Educators****:* Education of Machine Learning should focus on the whole lifecycle of Machine Learning development, including exploratory analysis with a focus on statistics, data analysis and data visualization. Moreover, practitioners with background on both data science and software engineering are a valuable resource for organizations. This emphasizes the importance of a transdisciplinary approach to AI education [31], [32] and it is congruent with previous work that reports that a Software Engineering mindset brings more awareness on the maintainability and stability of an AI project [20].

B. Threats to Validity

This subsection describes the threats and limitations of the study design and how these are mitigated. These limitations are categorized into researcher bias, respondent bias, interpretive validity, and generalizability, as reported by Maxwell [33] and Lincoln et al. [34].

1) ***Researcher Bias****:* Researcher bias is the threat that the results of the study are influenced by the knowledge and assumptions of the researchers, including the influence of the assumptions of the design, analysis, and sampling strategy.

A threat is introduced by the fact that participants are self-selected. This means that there might be employees in the company which should be included in the study but are not selected. During the planning phase, participants are selected with different roles and from different departments to have an as diverse starting point as possible. Thereafter, more participants are found by the recommendation of other interviewees and employees until we reach saturation on the information we get from the interviews, i.e. until no new information or viewpoint is gained from new subjects [35].

2) ***Respondent Bias****:* Respondent bias refers to the situation where respondents do not provide honest responses.

The results of the interviews rely on self-reported data. All people tend to judge the past disproportionately positive. This psychological phenomenon is known as rosy retrospection [36]. Furthermore, interviewees who know golden standards from for example literature may tell how things are supposed to be, in contrast with how they are in reality. These biases are mitigated by reassuring interviewees their answers will not be evaluated or judged and by asking them to think about a particular project they have been working on.

A methodological choice which can form a threat to validity is the fact that interviews are recorded. While the participants themselves permit the recording, they might be extra careful in giving risky statements on the record and therefore introduce bias in their answers. This threat is minimized by assuring the recordings themselves will not be published and all results which will be published are first approved by the corporate communication department.

3) ***Interpretive Validity****:* Interpretive validity concerns errors caused by wrongly interpreting participants' statements.

The interviews are processed by field-note taking and memoing. The primary threat to valid interpretation is imposing one's own meaning, instead of understanding the viewpoint of the participants and the meanings they attach to their words. To avoid these interpretation errors, the interviewers used open-ended follow-up questions which allowed the participant to elaborate on answers.

4) ***Generalizability****:* Generelizability refers to the extent to which one can extend the results to other settings than those directly studied.

This research is conducted in a large financial institution. Results may not seem generalizable to companies of much smaller size or different nature. A bank may be prone to more regulations than most companies and is dealing with more sensitive data. Still, every company has to comply with privacy regulations like the European GDPR. This suggests that results influenced by more strict regulations and compliance are just as relatable to other industries. Multiple case studies at organizations of different scale and nature are required for establishing more general results.

VII. Conclusions

The goal of this study is to understand the evolution of Machine Learning development and how state-of-the-art lifecycle models fit the current needs of the fintech industry. To that end, we conducted a case study with seventeen Machine Learning practitioners at the fintech company ING. Our key findings are: 1) CRISP-DM and TDSP are largely accurate; but 2) there are crucial steps missing from the fintech perspective, including feasibility study, documentation, model evaluation, and model monitoring; in particular, 3) the key challenges comprehend model governance and technology access.

Our research helps practitioners fine-tune their approach to Machine Learning development to fit fintech use cases. Additionally, it guides educators in defining learning objectives that meet the current needs in the industry. Finally, it paves the way for next research steps in reducing bottlenecks in the Machine Learning lifecycle, in particular study tool support for exploratory data analysis and data integration techniques, documentation, model governance, monitoring, and version control.

ACKNOWLEDGMENTS

The authors would like to thank Irene, Shiler, and Mieke, for their willing contributions to this project. The authors would also like to thank all the participants of the interviews at ING.

REFERENCES

[1] T. Menzies, "The five laws of se for ai," *IEEE Software*, vol. 37, no. 1, pp. 81–85, 2019.

[2] C. Mead and M. Ismail, "Analog vlsi implementation of neural systems," 1989.

[3] C.-J. Wu, D. Brooks, K. Chen, D. Chen, S. Choudhury, M. Dukhan, K. Hazelwood, E. Isaac, Y. Jia, B. Jia *et al.*, "Machine learning at facebook: Understanding inference at the edge," in *2019 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE, 2019, pp. 331–344.

[4] 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*. ACM, 2019, pp. 1743–1751.

[5] 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 *Proceedings of the 41st International Conference on Software Engineering: Software Engineering in Practice*. IEEE Press, 2019, pp. 291–300.

[6] C. Shearer, "The crisp-dm model: the new blueprint for data mining," *Journal of data warehousing*, vol. 5, no. 4, pp. 13–22, 2000.

[7] MicrosoftDocs. (2020) Team data science process documentation. [Online]. Available: https://docs.microsoft.com/en-us/azure/machine-learning/team-data-science-process/

[8] F. Martínez-Plumed, L. Contreras-Ochando, C. Ferri, J. H. Orallo, M. Kull, N. Lachiche, M. J. R. Quintana, and P. A. Flach, "Crisp-dm twenty years later: From data mining processes to data science trajectories," *IEEE Transactions on Knowledge and Data Engineering*, 2019.

[9] G. Mariscal, O. Marban, and C. Fernandez, "A survey of data mining and knowledge discovery process models and methodologies," *The Knowledge Engineering Review*, vol. 25, no. 2, pp. 137–166, 2010.

[10] S. Moyle and A. Jorge, "Ramsys-a methodology for supporting rapid remote collaborative data mining projects," in *ECML/PKDD01 Workshop: Integrating Aspects of Data Mining, Decision Support and Meta-learning (IDDM-2001)*, vol. 64, 2001.

[11] X. Wu, X. Zhu, G.-Q. Wu, and W. Ding, "Data mining with big data," *IEEE transactions on knowledge and data engineering*, vol. 26, no. 1, pp. 97–107, 2013.

[12] J. Rollins, "Foundational methodology for data science," *Domino Data Lab, Inc., Whitepaper*, 2015.

[13] F. Martínez-Plumed, L. Contreras-Ochando, C. Ferri, P. Flach, J. Hernández-Orallo, M. Kull, N. Lachiche, and M. J. Ramírez-Quintana, "Casp-dm: Context aware standard process for data mining," *arXiv preprint arXiv:1709.09003*, 2017.

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

[15] J. Lin and D. Ryaboy, "Scaling big data mining infrastructure: the twitter experience," *Acm SIGKDD Explorations Newsletter*, vol. 14, no. 2, pp. 6–19, 2013.

[16] M. Kim, T. Zimmermann, R. DeLine, and A. Begel, "Data scientists in software teams: State of the art and challenges," *IEEE Transactions on Software Engineering*, vol. 44, no. 11, pp. 1024–1038, 2017.

[17] Z. Ahmed, S. Amizadeh, M. Bilenko, R. Carr, W.-S. Chin, Y. Dekel, X. Dupre, V. Eksarevskiy, S. Filipi, T. Finley *et al.*, "Machine learning at microsoft with ml. net," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, 2019, pp. 2448–2458.

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

[19] E. Breck, S. Cai, E. Nielsen, M. Salib, and D. Sculley, "The ml test score: A rubric for ml production readiness and technical debt reduction," in *2017 IEEE International Conference on Big Data (Big Data)*. IEEE, 2017, pp. 1123–1132.

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

[21] R. K. Yin, *Case study research and applications: Design and methods*. Sage publications, 2017.

[22] P. Brereton, B. A. Kitchenham, D. Budgen, and Z. Li, "Using a protocol template for case study planning." in *EASE*, vol. 8. Citeseer, 2008, pp. 41–48.

[23] K.-J. Stol, P. Ralph, and B. Fitzgerald, "Grounded theory in software engineering research: a critical review and guidelines," in *Proceedings of the 38th International Conference on Software Engineering*, 2016, pp. 120–131.

[24] ING. (2019) Ing at a glance. [Online]. Available: https://www.ing.com/About-us/Profile/ING-at-a-glance.htm

[25] E. J. Halcomb and P. M. Davidson, "Is verbatim transcription of interview data always necessary?" *Applied nursing research*, vol. 19, no. 1, pp. 38–42, 2006.

[26] M. Haakman and L. Cruz, "Machine learning behind the scenes: An exploratory study in fintech - case study protocol," 2020.

[27] T. Wengraf, *Qualitative research interviewing: Biographic narrative and semi-structured methods*. Sage, 2001.

[28] F. A. Fasick, "Some uses of untranscribed tape recordings in survey research," *The Public Opinion Quarterly*, vol. 41, no. 4, pp. 549–552, 1977.

[29] I. . 2015, "Systems and software engineering–system life cycle processes," 2015.

[30] M. J. Ryan and L. S. Wheatcraft, "On the use of the terms verification and validation," in *INCOSE International Symposium*, vol. 27, no. 1. Wiley Online Library, 2017, pp. 1277–1290.

[31] Y. Wang, "Cognitive informatics: A new transdisciplinary research field," *Brain and Mind*, vol. 4, no. 2, pp. 115–127, 2003. [Online]. Available: https://doi.org/10.1023/A:1025419826662

[32] B. Nicolescu and A. Ertas, "Transdisciplinary theory and practice," *USA, TheATLAS*, 2008.

[33] J. Maxwell, "Understanding and validity in qualitative research," *Harvard educational review*, vol. 62, no. 3, pp. 279–301, 1992.

[34] Y. Lincoln and E. Guba, "Naturalistic inquiry," *Newbury Park, CA: SAGE.*, 1985.

[35] A. Strauss and J. Corbin, *Basics of qualitative research*. Sage publications, 1990.

[36] T. R. Mitchell, L. Thompson, E. Peterson, and R. Cronk, "Temporal adjustments in the evaluation of events: The "rosy view"," *Journal of experimental social psychology*, vol. 33, no. 4, pp. 421–448, 1997.