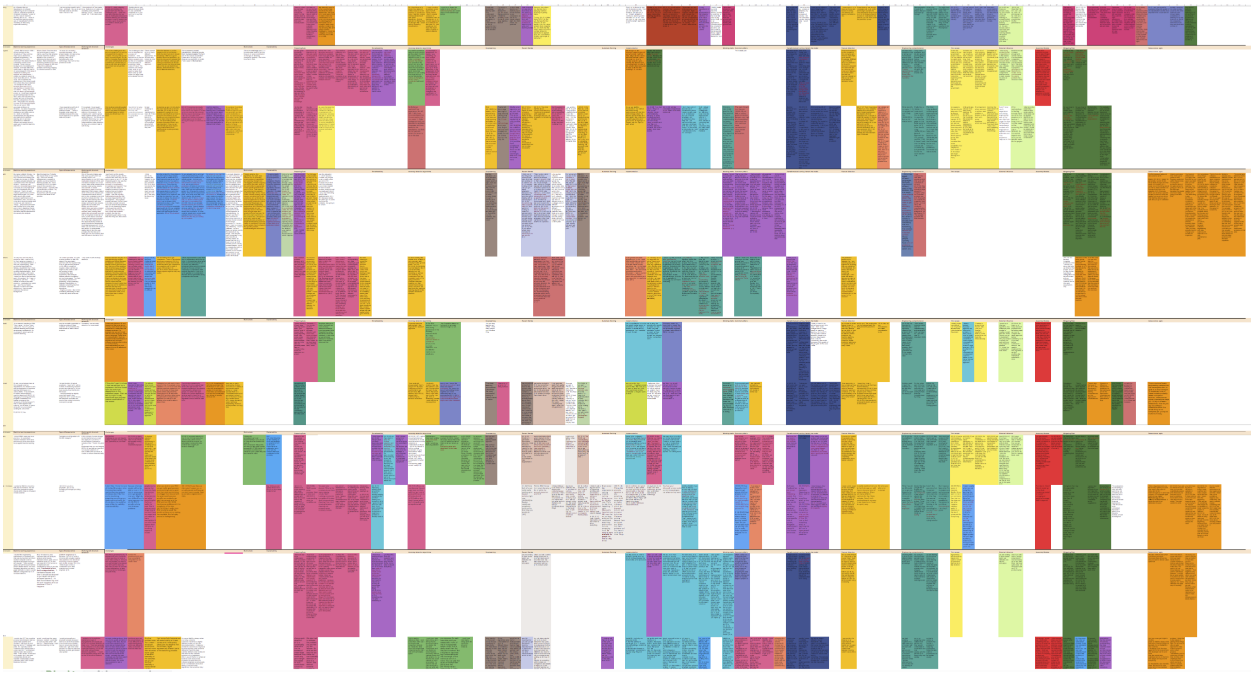


A Case Study of the Challenges with Applying Machine Learning in Industry

A Software Engineering Perspective

Master's thesis in Software Engineering and Technology

SAMUEL EKSMO & HANYAN LIU

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2019

Master's thesis 2019

A Case Study of the Challenges with Applying Machine Learning in Industry

A Software Engineering Perspective

SAMUEL EKSMO & HANYAN LIU

Department of Computer Science and Engineering
Chalmers University of Technology
University of Gothenburg
Gothenburg, Sweden 2019

A Case Study of the Challenges with Applying Machine Learning in Industry
A Software Engineering Perspective
SAMUEL EKSMO & HANYAN LIU

Supervisor: Regina Hebig, Software Engineering
Advisor: Johan Risberg, IBM
Examiner: Robert Feldt, Software Engineering

Master's Thesis 2019
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Color labeling method used in chapter 3.

Typeset in LaTeX
Gothenburg, Sweden 2019

A Case Study of the Challenges with Applying Machine Learning in Industry
A Software Engineering Perspective
SAMUEL EKSMO HANYAN LIU
Department of Computer Science and Engineering
Chalmers University of Technology and University of Gothenburg

Abstract

Data science is a growing trend and the advancement in machine learning and AI have been creating headlines in recent years. This has sparked an interest, not just in traditional IT-industries but also in businesses such as manufacturing, medicine and retail. Numerous industries are seeing potential in making their business more data driven and seeks to implement these trending technologies but few people know of the challenges that comes with applying it. This thesis aims at identifying the challenges, bridging the gap and lowering the entry barrier for engineers and researcher to contribute in the field of applied machine learning. In this case study, we examine how software engineers, data scientists and researchers can structure their work in order to increase the success rate of ML projects. Through interviews and a practical implementation test we analyze the underlying key concept that could help in bridging this gap. We conclude that software engineers can support in some initial data science activities, that communication between different stakeholders is crucial to the success of projects and that simpler ML models might be preferable in projects with time restrictions.

Keywords: Computer, science, computer science, engineering, project, thesis, Machine learning, software engineering, anomaly detection.

Acknowledgements

We would like to extend our appreciation to our supervisor, Regina Hebig at Chalmers University of Technology for providing support, structure and guidance from the beginning of the project to the very end.

We would like to extend our appreciation to our advisor Johan Risberg at IBM and fellow IBM'ers who shared their time and knowledge with us. We would also like to thank Energy Machines for letting us test their data and specifically David Brinnen for his enthusiasm and help with the project.

Samuel Eksmo Hanyan Liu, Gothenburg, May 2019

Contents

List of Figures

List of Tables

1

Introduction

Buzzwords[1] such as Machine Learning (ML) are hyped words and concepts that companies like to express in order to attract attention. However, ML and the Data Scientist profession are relatively new, and the industry is still evaluating how to use data analytics in an optimal way. Few people have actually been part of a ML project and know what ML really means and the hidden challenges that come with applying it (Black, 2019). One of these challenges revolves around the adaption process of ML. For example, how can software engineers apply and quickly deliver value to new customers when the training of high-quality ML models[2] requires extensive knowledge that is relatively rare (Allen, 2018).

This thesis revolves around a qualitative case study which aims to explore the process of adapting ML algorithms and models towards new data, new environments, and new software engineers. To be more specific, the focus was put on structuring the process of finding and adapting machine learning models towards new data, increasing the reusability of existing ML tools and adding a layer of support for software engineers, data scientists and researchers in the field of ML. Another part of the project is verifying the research result in a practical implementation process.

IBM is the main industry partner for this thesis and they see value in lowering the entry barrier for companies to use ML. One of their exiting projects has been to develop an ML anomaly detection tool that can be used in predictive maintenance[3]. They wish to research how ML tools can be adapted and quickly used within an optimal way out in the industry. This is relevant since there exists huge potential in the field of predictive maintenance (Joblogic, 2019). This thesis is written in collaboration with IBM who contribute with the foundation to the problem statement and a ML tool that was used together with an external client's (Energy Machines) data in order to test the findings of the thesis.

[1] Buzzword: "a word or expression of a particular subject area that has become fashionable by being used a lot, especially on television and in the newspapers:" (Cambride, 2019)

[2] Models are ML algorithms trained with data.

[3] Planned preventive maintenance is the traditional way of tackling potential faults in the maintenance field and is a billion dollar industry revolving around equipment replacements.

1.1 Research questions

This research focused on identifying the challenges with applying ML in industry. This was done by looking at ML adaption and challenges in the ML implementation process from multiple angles. The thesis also aimed at identifying possible solutions that could lower the entry barrier to implementing ML in the industry. This study clarifies these objectives through the following three research questions:

RQ1:

> *What are the necessary tasks in the processes of embedding a ML algorithm in a software system?*

RQ2:

> *What are the challenges in regards to new software environments, new data, new trends and new software engineers throughout the process of ML implementation in the industry?*

RQ3:

> *How do time scope and increasingly rapid technological advancement affect ML practitioners decisions and strategy during the process of adapting ML algorithms, models and tools?*[4]

There is a limited number of empirical studies that have conducted research related to bridging the gap between ML research and industry. A topic that has been researched even less is the software engineering aspect of the above-mentioned gap. Řehůřek (2017) summarizes some challenges between the two entities and describes the *mummy effect* which is the phenomenon where a research paper looks promising but once you try to implement the new technology you realize that corners were cut and everything crumbles apart.

This study aims to explore the gap between ML academic research and industry, increasing the understanding of challenges that software engineers, data scientists, and researchers need to face when looking to commercializing ML. Related research and important concepts such as the mummy effect are described more thoroughly in chapter 2.

To answer the above research questions, a qualitative case study was performed. Data was collected from a total of 16 semi-structured interviews with data scientists, researchers and ML practitioners from both industry and academia. The method process of a case study with interviews is further described in chapter 3 together with

[4] Time, money, news, and other external influences affect our everyday decisions. This research question aims to investigate how external parameters affect ML projects.

a method describing the practical verification process. In addition, the analysis was done through transcribing interviews and a color-labeling method which is described in detail in chapter 3. The results derived from the interviews in relation to the research questions are presented in chapter 4 and are structured according to their relevance. Furthermore, the results are discussed in chapter 5 and summarized in chapter 6

1.2 Delimitations

This thesis has been done in collaboration with IBM and Energy Machines. Hence, for safety reasons, details concerning their intellectual property rights have been left out of the report. Furthermore, time restrictions limited the scope of the thesis, e.g. what type of interviewees that was possible to interview. Early in the process, it was decided to focus on data scientists and researcher in order to collect experience relevant for ML projects as opposed to software engineers with ML experience, since the later proved to be very hard to find.

The population interviewed where mainly data scientist from IBM. It would be interesting to include more people from other companies in order to possibly make the findings more general. However, focusing on IBM significantly simplified the search for interviewees, hence it was an active decision that was needed in order to fit the time restrictions.

In the planning phase, it was determined that the analysis of the case study would be done semi-parallel to the practical verification. This was done in order to fit the time scope but it may also have influenced certain choices and tools used in the practical verification. The main idea behind the implementation test was to test the findings but it may have introduced an amount of bias since it is largely a self-reporting experience.

2

Background

In the following section, related research and important concepts will be is presented. It starts with the basic principles behind machine learning, anomaly detection, roles of data scientist and moves deeper into the existing gap and the challenges. Furthermore, it continues with transformation and reusability.

2.1 Machine Learning

The quest for finding patterns in data is an old one according to Bishop (2006). For example, Johannes Kepler discovered the empirical laws of planetary motions through pattern recognition and these findings laid the foundation for the modern world we know of today. Machine learning is fundamentally about the automatic discovery of regularities in data through computer-based algorithms, in other words, pattern recognition through automatic processing of data by evolving algorithms.

The author further explains that machine learning is also about probability theory where uncertainty plays a key role. Given a fixed size of data, you are limited by the information at hand since the model can only make predictions based on available data.

Adding complex noise in the data that could be derived from non exact measurements, the prediction becomes further probabilist. In machine learning, this uncertainty is quantified and manipulated and together with decision theory, it creates a powerful tool that lets people make an optimal prediction based on the gathered information. Even though there may be noise and incomplete data.

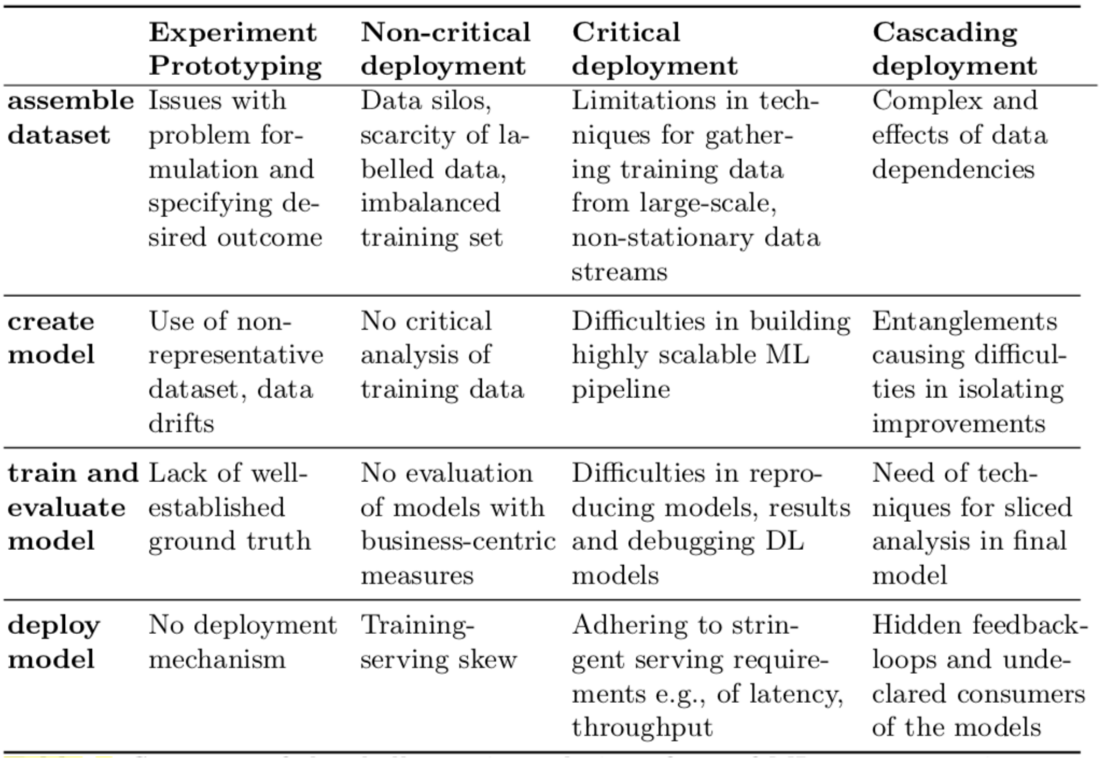
2.1.1 The Machine Learning process

According to Sapp (2017), the first step of applying machine learning to a system is understanding the problem. A problem taxonomy, i.e classification, is one way to start structuring a problem. These classifications can be; exploring and determining new patterns in data or clustering; predictive ML; unsupervised ML learning when there is no prior knowledge of the output and supervised learning, when labeled training data is available. The next step involves a data preparation stage, identifying where data supports the problem and cleaning the data for ML execution, including data transformation, normalization, missing values, etc.

Sapp (2017) further describes the process as modeling, validation, and execution. These steps involve considering computer resources, numbers of features, computation time and criteria such as scalability, reliability, and efficiency. A range of ML algorithms is scouted for training the model in this phase. Data is changeable, so the most flexible algorithms are usually preferred. During an iterative execution, the performance of the models is measured for validation. It is important to track metrics of deployed algorithms periodically in order to properly measure the accuracy of models, i.e. in order to determine any seasonality. For the final deployment stage, the outcome of models will inform decisions, feed applications or be stored for future analysis. Models need to be updated and retrained in order to maintain the quality and accuracy of prediction.

2.1.2 Different stages of ML

Lwakatare, Raj, Bosch, Holmström Olsson, and Crnkovic (2019) describes four different stages of ML systems and summarizes the challenges of each stage, see Figure 2.1. Each deployment iteration faces special challenges unique for that setting. For example, in the prototyping stage, it can be hard to accurately specify a correct problem formulation. While in a critical deployment there instead may arise problems related to scalability. It is clear that each step has its own specific threats that may not be clear from the beginning. Instead, each ML project needs to face new challenges as the model and surrounding environment progresses.



	Experiment Prototyping	Non-critical deployment	Critical deployment	Cascading deployment
assemble dataset	Issues with problem formulation and specifying desired outcome	Data silos, scarcity of labelled data, imbalanced training set	Limitations in techniques for gathering training data from large-scale, non-stationary data streams	Complex and effects of data dependencies
create model	Use of non-representative dataset, data drifts	No critical analysis of training data	Difficulties in building highly scalable ML pipeline	Entanglements causing difficulties in isolating improvements
train and evaluate model	Lack of well-established ground truth	No evaluation of models with business-centric measures	Difficulties in reproducing models, results and debugging DL models	Need of techniques for sliced analysis in final model
deploy model	No deployment mechanism	Training-serving skew	Adhering to stringent serving requirements e.g., of latency, throughput	Hidden feedback-loops and undeclared consumers of the models

Figure 2.1: Challenges in different stages of ML systems

2.2 Challenges in Machine Learning

In this section, existing challenges were listed from the literature including long term costs in machine learning fast development – hidden technical debt, possible challenges for commercial interests, problems related to huge data management for training in ML and challenges in Deep Learning.

2.2.1 Hidden Technical debt in Machine Learning

Sculley et al. (2015) argues that ML systems have a high tendency to acquire technical debt[1] since ML systems require all the traditional maintenance problems together with all the unique challenges of ML. This involves issues at the system level which is a higher abstraction level than at the code level. For example, a system might subtly evolve differently depending on user data, introducing small corruptions that gradually reduce intended structure and performance. Refactoring and other traditional tools that are used to tackle hidden technical debt are not enough to counter these new, higher level challenges.

[1] Technical debt, first introduced by Ward Cunningham in 1992 refers to the concept of accumulating long term costs as a consequence of moving fast in Software Engineering

The authors further explain trade-offs and technical debt specific to ML systems that are accumulating in the ML community. Bellow is a short summary list of some of the challenges.

- Correction cascades can occur in a situation when it is tempting to reuse a model that was intended for a specific problem A to solve a slightly different problem A*. Using the first model as prior and quickly correct the model with the new data. This may work smoothly and beneficially but it also increases the correct gap for the next problem. Say this the scenario reoccurs five times, then the original model may be vastly different from the fifth, cascading the correction gap.

- Data dependencies are necessary in order to create ML systems but they are also more costly than code dependencies since they are harder to detect. Examples of data dependencies that cause a disturbance are unstable data flow from sensors. Signals that change their behavior over time such as mean temperature in the winter half-year compared to summer. Furthermore, there can be data dependencies in bundled features that together evaluate to be useful but was the rushed decision in order to meet a deadline. Resulting in features that may contribute little or no value. One way to tackle underutilized features is to conduct the leave-one-feature-out evaluation.

- Feedback loops and scalability is also a reoccurring challenge where some algorithms and settings in smaller pilot studies do not scale well to the real-world problem. There may also be hidden feedback loops that cause the model to take misguided action, one example being two stock-market prediction models that trigger buy or sell on each others' outputs.

- Glue code is the additional code required to adopt a generic ML package or library to your own solution. The amount of supporting code that is needed to get the right input and output to and from the model. It might be cheaper and more efficient to create a customized solution from the beginning.

- Pipeline jungles is another phenomenon that may arise in the data preparation phase. New signals may lead to a bundle of different supporting structures that together becomes extremely complex to manage and needs extensive end-to-end testing. To counter this it is important that engineers and researchers work closely together.

- Other smells worth mentioning are multiple language smell, prototype smell, prediction bias, and legacy features.

2.2.2 Conflict between ML research and commercial interests

Dominique Foray (2009) argue that commercial interests may mitigate combined research efforts. Industry to a large degree relies on intellectual property rights (ITP) developed by the researcher. The lock-in mechanism may discourage other researchers from building upon this research in order to advance the field. Hence, there is a dilemma between commercial interest and scientific advancement.

The author further describes that commercial interest may affect some research more than others. For example, government research laboratories (GRLs) and research universities (RUs) are two main institutions for research. GRLs research is organized to targeted objectives and naturally, is not as "free" as in RU, while RU researches are carried individually, the possibility of research exhibiting hysteresis effects[2] is then typically larger at the later.

2.2.3 The Mummy effect

Řehůřek (2017) states that the Mummy effect is an increasingly reoccurring phenomenon in the ML field. As mentioned in the introduction, the Mummy effect occurs when research articles brag about revolutionary findings that quickly falls apart during external reenactments due to poor methods. Like an Egyptian mummy that looks intact but breaks at a light touch. The phenomenon is also applicable in an industrial context since multiple companies may get drawn into trying a technology that is not transferable or adaptable to their portfolio.

Jarmul (2017) exemplifies this in her talk at the PyData Conference in 2017 by describing some challenges with deep learning and why it may be better to stick to simpler, more reliable models. One reason is that the complexity of the deep learning models may affect the interpretability of models. If a researcher cannot explain how the model works, then it may lead to other problems such as reproducibility. Sticking to Simpler models that the researcher can understand and argue for may be a better way forth for the community surrounding ML and AI.

2.2.4 Important aspects in ML adaption

López, Mordvanyuk, Gay, and Pla (2018) describe the challenges in transforming sensor data to useful information that can be used in clinical findings. The process

[2] hysteresis effect: researchers reputation increases the chance of receiving research grants which further spirals reputation and new grants. Which may mitigate the system's capability of identifying the best research (Foray, 2004).

of interpreting sensor data in their specific research requires expertise from nurses and doctors. The author further states that AI an ML models can play an important role in supporting the interpretation process. However, he also stresses that knowledgeable people are required who can make the decisions regarding suitable representations of data, what ML models to use in the vast selection of alternatives, the wanted outcome, and quality of findings. The above challenges are relevant and applicable also in IBM's case since a qualified operator is needed in order to determine if a sensor is really abnormal[3], if the data representation is useful and how the alarm should be treated.

Menzies and Rogers (2015) describe some challenges with working with data, for example, not all data from a client will be useful. You need to set up some relevance filter that can help with sorting the useful data from the corrupt data. To have a successful data science project there also needs to be (1) light on the users, not just the algorithms, (2) the math and tools and problem domain should be well known, (3) inspect and clean data before use, (4) Data science is an iterative process that usually means that the prediction needs to be tested several times before it can show useful results.

2.2.5 Challenges of Data Management in ML Production

Polyzotis, Roy, Whang, and Zinkevich (2017) aim to increase the awareness of the data management challenges in regards to analyzing, modeling, validating, enriching and debugging data in an industrial setting. The authors specifically highlight the challenges in building robust data pipelines for ML in the industry. For example, one problem described by the authors is making sure the data passed through the pipeline is validated, in other words, that no invalid data is used that may affect the model negatively or in later stages corrupt other reliant programs. Another problem is that the countermeasures to deal with invalidated data such as enforcing a specific format on the data is not always usable in a production setting.

Polyzotis et al. (2017) further describes that engineers and data scientist needs to spend a big portion of their time understanding the raw data before it can be deployed. Common methods used are different visual plots, the range of features, correlations and removing outliers. This can be hugely time-consuming, especially when up-scaled models require large amounts of data. Requiring domain knowledge and mapping the external dependencies is also necessary in order to produce any significant result.

Another important aspect in the preparation phase, related to developing a data pipeline is the cleaning of the data. Polyzotis et al. (2017) describes the challenging

[3] IBM has chosen not to classify anomalies in detail, instead of the tool output the sensor ids that are alerting the system. They are depending on operators to make the final judgment of the systems state.

process of cleaning a data pipeline, i.e evaluating a set of data. This is organized in three steps, the first being: identifying where an error has occurred. If the distribution of a parameter has changed over time, it is crucial to determine the surrounding contributing factors. Is it a naturally occurring phenomenon in the data or is it a real error. Localizing the start and stop for the occurring change. Next is measuring the impact of the error. Sometimes it can be preferred to ignore the error-prone data signal if it does not significantly affect the overall performance. However, in order to measure the impact, it is necessary to run a test on the data as described by López et al. (2018). The third step is to remove the error which can be done by finding the bug or more easily updating the pipeline to not include the data tag.

Polyzotis et al. (2017) also explains a few approaches to enriching the data. Enriching meaning to increase the size of the training data, exploring new features and using transformations of data in order to improve the model. Designing a data catalog with responding correlations might help in this task and the author also stresses that the team needs to understand the effect of the enrichment in order asses it. Finally, the authors also describes the challenge of sensitive data (etc hospital journals) in terms of excessive overhead. The administration might slow down the development process, hence one possible action could be to approximate the size and quality of the data without additional access before it is tested.

2.2.6 Challenges in Deep Learning

Arpteg, Brinne, Crnkovic-Friis, and Bosch (2018) analyzed 7 deep learning projects and identified 12 challenges. These challenges were then further categorized into 3 areas: development, production, organization. Below is a short summation of each area.

- **Development:** A large number of experiments need to be performed to identify the optimal model, so the exact version of components like hardware, platform, configuration, training data need to be documented. This need version control tools for DL. Another transparency problem is that in DL, it's difficult to isolate functional areas. If the model is complex and inherently irreducible, the explanation would be complex too. In addition, using framework and libraries make it difficult to debug problems and test the data. Manual evaluation is impossible due to millions of parameters.

- **Production:** The training time for DL is few days up to several weeks while the platforms update weekly or daily with noticeable improvements. In a production environment, it may be increasingly important to maximize performance in accuracy, this may lead to a conflict in speed. In addition, it is hard to cover all edge cases in production. Models in a big data context can make the prediction into a self-fulfilling prophecy. For example, a prediction about a students university acceptance rate based on the students' environment may discourage a student from applying to a certain university. Another problem is the glue code and supporting system. Not keeping those up to date can introduce new challenges in production.

- **Organization:** The collaboration between different teams is a challenge. For the ML project, it is unclear for the model could achieve the goal and hard to decrease the scope and do effort estimation. From a privacy and data safety perspective, it is hard for a designer to control where and how the big data is stored. No comprehensive terms of service agreements from companies may introduce a question of data safety and privacy. Another aspect is the cultural difference between data scientists and engineers, where a data scientist cares more about the prediction result and engineer care more about maintainability and stability.

2.3 Roles in Machine Learning

Software engineers and data scientists have different tasks in ML processes. In this section, the workflow of software engineers and the roles of data scientists are described.

2.3.1 Software Engineering for Machine Learning

Saleema Amershi (2019) describes a workflow consisting of nine activities that every data scientist goes through, see Figure 2.2 and presents a process maturity metric that can help teams orient themselves in their AI development. The earlier stages such as data accumulation and cleaning seem to be the most challenging for all respondents while finding the education is ranked as harder by more junior experienced respondents while more experienced respondents rank model-evaluation as more challenging etc. In the case study, they interviewed 114 stakeholders with ML experience and later conducted a survey based on the results. In the study, they also describe that some teams interviewed had developed their own best practices to help in developing scalable AI applications which included combining some of the workflow steps, automation of data streaming, agile workflows and modularity. They also present three main differences in AI development and traditional software engineering. The first being that the initial steps, i.e. discovering, managing, and version control of data is more complex than corresponding activities in software development. Secondly, that scaling and customization of models require ML expertise to be involved in the process to a certain degree, pure software engineering is not enough. Thirdly, that modularity is more complex when working with AI since models and data pipelines usually entangle. These findings can help to solve challenges that may arise when creating large scale AI solutions.

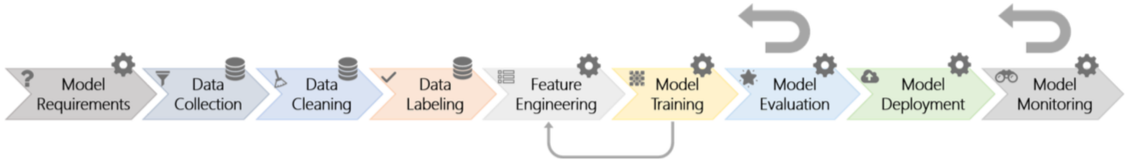


Figure 2.2: ML workflow process presented by Saleema Amershi (2019).

2.3.2 Data scientist roles

Kim et al. (2016) investigates the different roles a Data scientist can have in software engineering. Only recently have companies started to understand the power of data-driven decisions. Gathering and cultivating data is a growing trend but it also requires competent engineers or data scientists. The young profession comes with a variety of meanings for different people in the industry. After a series of interviews with data scientists, the authors summarized the different roles into five types, see Table 2.1.

Role	Explanation
Insight Providers	work with engineers to collect the data needed to inform decisions that managers make
Modeling Specialists	use their machine learning expertise to build predictive models
Platform Builders	create data platforms, balancing both engineering and data analysis concerns
Polymaths	do all data science activities themselves
Team Leaders	run teams of data scientists and spread best practices

Table 2.1: The five main different data scientist roles as presented by Kim et al. (2016).

2.4 ML Technologies

This section covers some of the technologies relevant for data scientists working with implementing ML in industry. It also has a section describing different types of anomalies in anomaly detection.

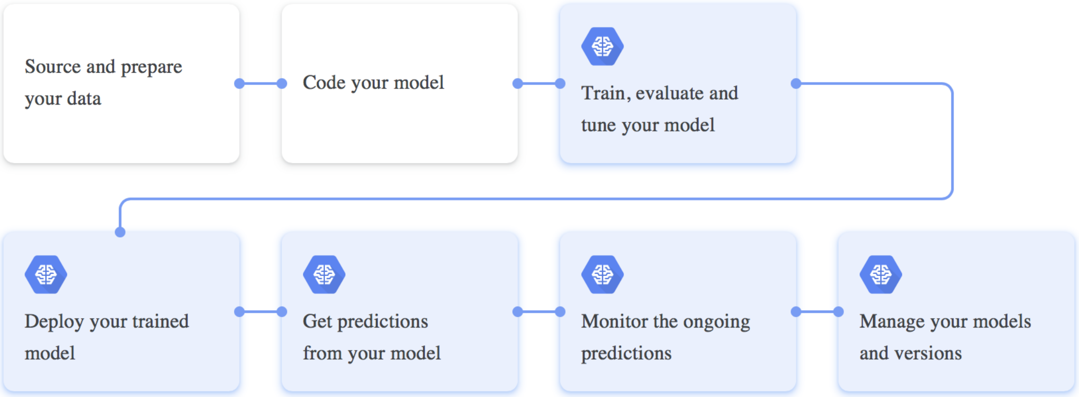

Source and prepare your data
Code your model
Train, evaluate and tune your model
Deploy your trained model
Get predictions from your model
Monitor the ongoing predictions
Manage your models and versions

Figure 2.3: Workflow process in Google clouds AI platform.

2.4.1 AI-platforms

Xing et al. (2015) investigates the problem of developing distributed machine learning frameworks, e.g. ML platforms, in terms of efficiency, correctness, programmability, and tradeoffs. AI platforms are a recent trend that aims to provide space for scalable ML models as a result of single machine bottlenecks that have a hard time processing Big Data[4]. Many platforms provide partial solutions to making ML more accessible but fail to address all problems. Xing et al. (2015) further aim to describe these differences and present a new distributed ML framework (Petuum) that specializes in ML-centric optimization-theoretic principle instead of the more conventional operational objectives centered platforms. Google's Tensorflow is another example of a ML platform (Tensorflow, 2019). According to Xing et al. (2015), AI platforms are one area being developed in order to bridge the gap between research and production. Figure 2.3 is one example of an AI-platform flow chart.

2.4.2 Narrow AI to Broad AI

Russell and Norvig (2009) describe Artificial Narrow Intelligence (Narrow AI) as an AI that is programmed to be used for a single task such as predicting the weather. Narrow AI trains a model for a specific problem and from a predetermined data set. In order to make the Narrow AI more similar to human intelligence, i.e. consciousness, Narrow AI needs the ability to transfer the learned skills. In other words, the Narrow AI needs to be able to adapt to new information and objectives. This concept is proposed to be named Broad AI or Artificial General Intelligence(General AI). General AI uses multiple data streams in order to identify a specific situation and make agile cognitive complex decisions that in the end can produce better results. In short, all "AI" that is used today is Narrow AI but the author predicts Broad AI will eventually be developed to some degree.

[4] Big Data: Large sets of data e.g petabytes, terabytes of data

2.4.3 Anomaly detection

Since this thesis also involves a practical implementation surrounding anomaly detection it is relevant to elaborate on some of the concepts.

Chandola, Banerjee, and Kumar (2009) describes anomaly detection as the problem of identifying patterns in data that do not reflect expected behavior. Nonconforming patterns are generally referred to as anomalies but different domains may use other terms such as outliers, exceptions, surprises, aberrations or contaminants. The authors further explain that anomaly detection can be used in a verity of application fields such as fraud detection in credit cards, health care, intrusion detection and fault detection in safety-critical systems. The last being the critical focus point in this thesis.

According to Chandola et al. (2009) anomalies can be divided into three different categories:

Point anomalies, individual points are considered anomalies if they fluctuate compared to the rest of the data. It is the simplest and most used category in anomaly detection research. For example, in credit card fraud detection, it might be a person's transactions that are subject to modeling. If a transaction involves a high sum in relation to the usual transactions it might be termed abnormal.

Contextual anomalies is when a data point might be out of place in a certain context but not when compared to the whole data set. It would then be termed as a contextual anomaly. This approach requires two sets of attributes. First, a contextual attribute which determines the context of the point and behavior attributes that represents the noncontextual characteristics of the data point.

Collective anomalies, if a group of points is behaving in an non-typical way compared to the rest of the data set, they might be termed collective anomalies. So individual points of a group might not be anomalies by themselves but as a group, they correspond to abnormal behavior.

3

Methods

The research design and process is described chronologically and follows a case study design and its corresponding structure.

The process of adapting ML algorithms and models is a contemporary phenomenon occurring all over the world in different settings. According to Runeson and Höst (2009) a **case study** lets researchers look at the phenomenon in their natural, real-life context. This fits the purpose and research questions of this thesis, *exploring how researchers and practitioners are handling the gap between ML research and industry.* The factors in case studies cannot be controlled, instead, researchers can look at the phenomenon interacting with the context it is in. The research findings are obtained by looking at analytic depth in typical and special cases. This case study was carried out based on what Runeson and Höst (2009) and Lenberg, Feldt, Wallgren Tengberb, Tidefors, and Graziotin (2017)'s description in their articles, the flow of our methods is shown in Figure 3.1.

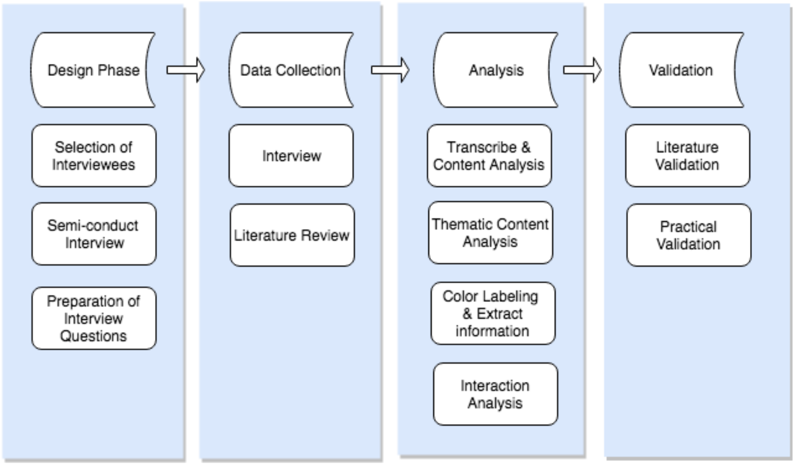


Figure 3.1: Method process

3.1 Design phase

According to Runeson and Höst (2009), the design phase defines the structure of a case study into a work plan. The plan is also built upon juridical, professional and ethical requirements of the thesis. In our case, Chalmers', IBM's and Energy Machines' ideas were taken into account since they were all stakeholders in this thesis. All parties had different value driven requests that needed to be addressed before the actual work plan could be executed. Consequently, each RQ was developed to incorporate a part of all the different stakeholder's requirements, e.g. for Chalmers, this mainly revolved around making a novel contribution to the Software Engineering field. After a few iterations of the proposal, the objective of the project was divided into two separate parts: the first being a qualitative study and the second being a practical verification of the findings from the first study.

Runeson and Höst (2009) describe steps that can help researchers in their case study process. These steps can be used in order to help validate the research progression. For example, the author emphasizes that the initial steps of a case study are depending on a thorough literature review in order to formulate good research questions. In order to formulate relevant research questions, the theory for this thesis was organized by searching for topics related to software engineering and ML, anomaly detection, challenges in ML, etc. Literature was chosen based on publication date, number of citations, and type of study primarily. After the first draft of the thesis proposal and corresponding feedback, the search pivoted to look for more literature regarding the software engineering role in ML implementation processes. The amount of literature proved to be scarce but after a thorough search and some tips about new research papers (about to be published), the combined theory was deemed sufficient. From the theory and through dialog with the stakeholders, the identified gap in the ML field and the scope of the thesis was decided to revolve around; how to get ML out in the industry from a software engineering perspective.

3.1.1 Interviewees

The interviewees were mainly selected based on their experience and development of ML. Both researchers, data science practitioners in the industry (IBM, Ericsson) and in university (Chalmers University of Technology) with senior or junior experience were interviewed in order to get a wide spectrum of input. At the start of each interview, the interviewees were asked to place themselves among five different types of data scientist roles. According to Kim et al. (2016) there are five different types of data scientist roles which are explained in subsection 2.3.2. This question is highlighted for getting a clearer view of the interviewees owns perception of their working role. In total, 16 interviews were conducted, Table 3.1 specifies all interviews, including the interviewee's title, experience, and role type according to the classification described by Kim et al. (2016)

ID	ML Experience	Title	Type	Gender	From	Working Area
1	15 years	Senior researcher	Team leader, Polymath	male	Europe	Academia
2	2 years	Postdoc	Insight provider	female	Europe	Academia
3	3 years	Phd	Polymath	male	Asia	Academia
4	20 years	Senior data scientist	Modeling specialist	male	Asia	Industry
5	5 years	Senior researcher	Platform builder	male	Europe	Academia
6	15 years	Senior researcher	Team leader	male	North America	Industry
7	2 years	Junior data scientist	Data specialist	female	Europe	Industry
8	10 years	Senior data scientist	Modeling specialist Insight provider Team leader	male	Asia	Industry
9	10 years	Senior researcher	Team leader	male	Europe	Academia
10	6 years	Senior data Scientist	Team leader	female	Europe	Industry
11	2 years	Junior data scientist	Insight provider	male	Europe	Industry
12	2 years	Junior data scientist	Polymath	male	Europe	Industry
13	2 years	Junior data scientist	Platform builder	female	Europe	Industry
14	5 years	Senior data scientist	Polymath	male	Europe	Industry
15	5 years	Data scientist	Team leader	female	Europe	Industry
16	15 years	Senior researcher	Modeling specialist	male	Asia	Industry

Table 3.1: The participants interviewed for this study.

3.1.2 Semi-structured Interview

According to Runeson, Host, Rainer, and Regnell (2012), three types of interview formats can be chosen when conducting a case study: fully structured interviews, semi-structured interviews or unstructured interviews. Since we did not know beforehand how homogeneous the interviewees would be, and we wanted to have the freedom of adjusting the questions towards each individual subject, we choose to design a semi-structured interview protocol with a mix of open and closed questions. Semi-structured interviews let the interviewer adapt the questions to a larger degree than fully structured interviews but at the same time keeps a clear structure.

3.1.3 Interview design and iteration

The funnel principle of structuring interview protocol described by Runeson et al. (2012) was selected, which starts broad with warm-up questions and then slims down to targeted questions. Therefore, the first set of interview questions was designed in parallel to the first thesis proposal, including 11 basic questions based on our research questions. Furthermore, after some feedback from our supervisor, we structured the interview protocol to include more in-depth questions. After the third iteration, we conducted a pilot interview with a Ph.D. student in Machine Learning and afterward adjusted some questions in the list based on the perceived flow of the interview. The result from the pilot interview provided insights, so we decided to keep the pilot interview and analyzed it together with the other 15 interviews. The final set of interview questions can be found in Appendix A.

As stated in subsection 3.1.2, the semi-structured interview allows for tweaking of questions to better reflect the background of the interviewee. In addition, the time limit played a role in prioritizing of questions during some interviews.

3.2 Data collection

Data collection included identifying, contacting, scheduling and conducting interviews. The second phase proved to be more time consuming and challenging than first anticipated since it was hard to access to people with ML experience. However, our IBM supervisor provided a list of relevant IBM people of which the majority agreed to participate in this research.

A total of 16 interviews were conducted due to the saturation of new insights. The interview duration ranged from 35 to 65 minutes and all interviews were recorded with the interviewee's approval. Three or four interviews were scheduled every week.

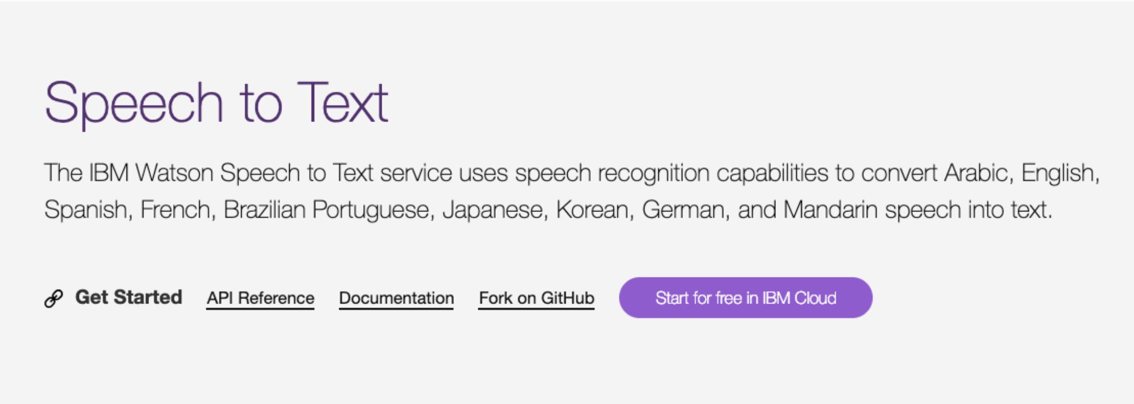
3.3 Analysis of interview data

In order to analyze qualitative data in a case study setting, Runeson et al. (2012) and Vaismoradi, Turunen, and Bondas (2013) suggest that results should be derived through theme identification. Themes, patterns, and relationships between the answers of interviewees are labeled and later summarized in a structured manner. Following subsections describe how the analysis was conducted as suggested by Runeson et al. (2012) and Vaismoradi et al. (2013).

3.3.1 Analysis of transcripts and interview content

Runeson et al. (2012) build their analysis method on transcripts of interviews. This approach was taken in this study, and all interviewees were asked prior to the interview if they could be recorded. As soon as possible after each interview, transcribing was conducted and in total it summarized to 128 pages or 78072 words. This made it possible for both researchers to listen actively during the interviews without the need for extensive note-taking. According to Guion, Diehl, and McDonald (2001), active listening facilitates relevant follow-up questions which is one of the main advantages of semi-structured interviews.

Transcribing interviews is time-consuming but it makes interviewers feel comfortable that all questions and answers are covered and that the accuracy for a future result is solid. However, some words and sentences were in some cases labeled as *unheard* due to different accents combined with blurry audio. Furthermore, in rare cases, the interviewee also asked not to be cited in certain areas that might be subject to confidentiality. These paragraphs were left out of the transcriptions. According to Bailey (2008) there is also a question of the level of detail that should be covered in the transcription. The level was kept high by for example including relevant moods such as laughter in the transcripts in order to more closely reflect the true meaning of the answers.

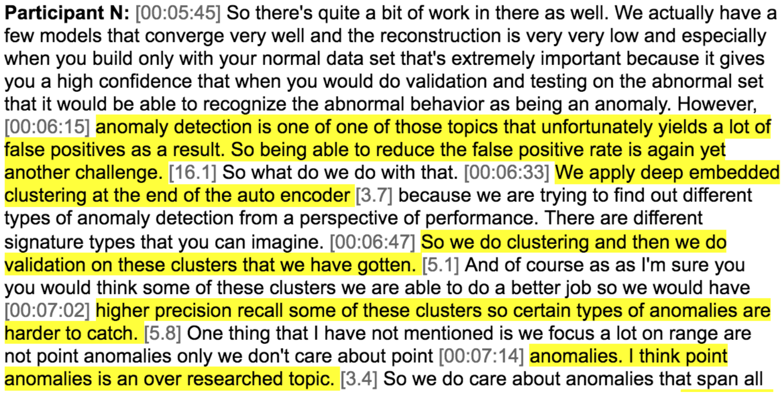


Speech to Text

The IBM Watson Speech to Text service uses speech recognition capabilities to convert Arabic, English, Spanish, French, Brazilian Portuguese, Japanese, Korean, German, and Mandarin speech into text.

Figure 3.2: IBM Watson natural language processor API used in the transcribing process

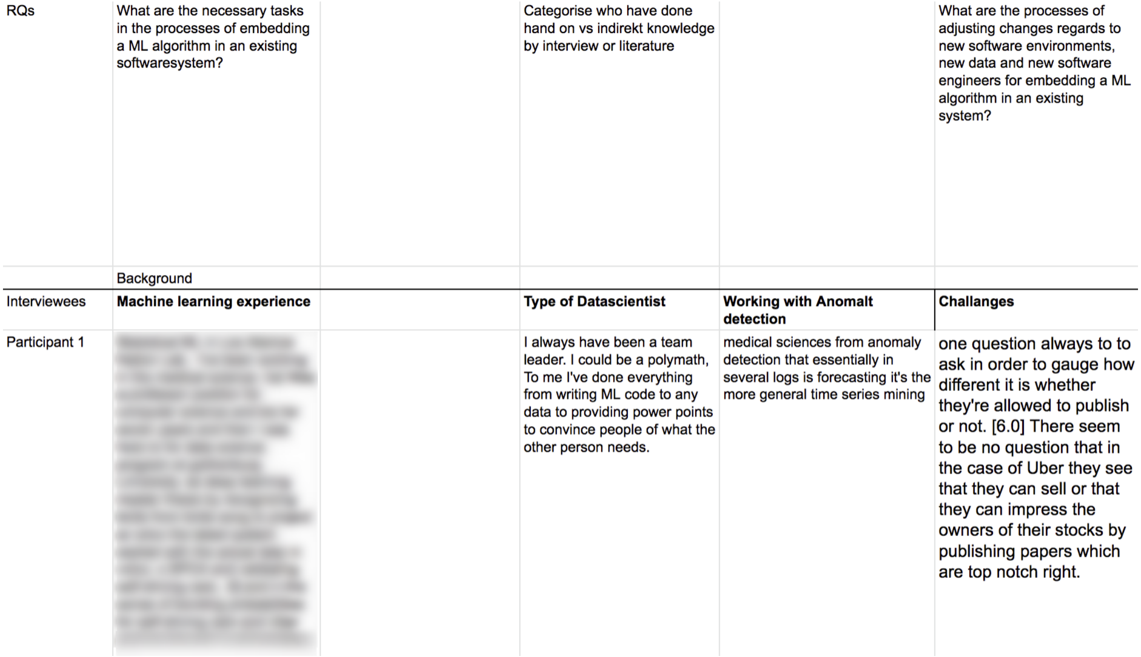
According to Runeson et al. (2012) transcribing interviews also increase the researcher's comprehension of the qualitative data and facilitates in-depth analysis since the researchers are required to listen to the recording multiple times. In this research process, we explored three online Natural Language Processors (NLP), one being IBM's own service shown in Figure 3.2, these were tested to conduct the test edition of transcribed results. The NLP service we finally used generated a draft text document which in the majority of the cases saved considerate amounts of time and energy.



Participant N: [00:05:45] So there's quite a bit of work in there as well. We actually have a few models that converge very well and the reconstruction is very very low and especially when you build only with your normal data set that's extremely important because it gives you a high confidence that when you would do validation and testing on the abnormal set that it would be able to recognize the abnormal behavior as being an anomaly. However, [00:06:15] anomaly detection is one of one of those topics that unfortunately yields a lot of false positives as a result. So being able to reduce the false positive rate is again yet another challenge. [16.1] So what do we do with that. [00:06:33] We apply deep embedded clustering at the end of the auto encoder [3.7] because we are trying to find out different types of anomaly detection from a perspective of performance. There are different signature types that you can imagine. [00:06:47] So we do clustering and then we do validation on these clusters that we have gotten. [5.1] And of course as as I'm sure you you would think some of these clusters we are able to do a better job so we would have [00:07:02] higher precision recall some of these clusters so certain types of anomalies are harder to catch. [5.8] One thing that I have not mentioned is we focus a lot on range are not point anomalies only we don't care about point [00:07:14] anomalies. I think point anomalies is an over researched topic. [3.4] So we do care about anomalies that span all

Figure 3.3: Highlighted example results from the inital analysis.

However, all transcripts required additional manual adjustments. In cases where interviewees had strong accents, the NLP tools were not sufficient, therefore, we had to manually transcribe those interviews. During the coding of transcripts, we also conducted the first analysis of the data shown as an example in Figure 3.3. While listening to the recordings and adjusting the transcript drafts from the NLP tool(s), answers relevant to the RQs and other noteworthy paragraphs were highlighted by us. This constitutes the initial high-level labeling of relevant findings.



RQs	What are the necessary tasks in the processes of embedding a ML algorithm in an existing softwaresystem?		Categorise who have done hand on vs indirekt knowledge by interview or literature		What are the processes of adjusting changes regards to new software environments, new data and new software engineers for embedding a ML algorithm in an existing system?
	Background				
Interviewees	**Machine learning experience**		**Type of Datascientist**	**Working with Anomalt detection**	**Challanges**
Participant 1	[redacted]		I always have been a team leader. I could be a polymath, To me I've done everything from writing ML code to any data to providing power points to convince people of what the other person needs.	medical sciences from anomaly detection that essentially in several logs is forecasting it's the more general time series mining	one question always to to ask in order to gauge how different it is whether they're allowed to publish or not. [6.0] There seem to be no question that in the case of Uber they see that they can sell or that they can impress the owners of their stocks by publishing papers which are top notch right.

Figure 3.4: Theme extract example - columns where used to sort the result

3.3.2 Thematic content analysis

As shown in Figure 3.4, we extracted themes out of the transcribed interview texts in regards to our three research questions, putting all valuable transcribed content into a large excel table. The ML experience information of the participant is blurred to protect the privacy of the participant, the research questions are put in the first row, participant number and name are put in the first column. Each relevant theme got a growing number of columns depending on the number of insightful answers. The whole table is large and to give an idea about it, part of the different stages are shown in Figure 3.5.

Figure 3.5: Part of theme table with citations related to different themes.

The themes were reorganized in the color labeling stage. Besides, the explanation of final edition theme table is shown in chapter 4 in Table 4.1.

3.3.3 Color Labeling and information extraction

After the two initial iterations of theme identification: one is described in subsection 3.3.2, another is what we did in the color labeling stage, in the next step, the citations in each box were carefully inspected and highlighted with an individual headline and theme color, shown in Figure 3.6. Some citation blocks spanned over multiple themes. These were given the theme that reflected the specific questions the most. All boxes got color labels with the exception of some irrelevant answers and warm up questions. The headlines and respective theme colors are listed in Figure 3.7. Color labeling helped us to investigate the theme in more detailed, with new valuable themes were discovered.

While color-labeling the data, the citations were summarized into bullet points in order to get a more compact collection of statements from every theme. These bullet points were analyzed, combined and restructured in order to distinguish deeper insights. The description of themes and bullet points in Figure 3.7 is presented later in chapter 4.

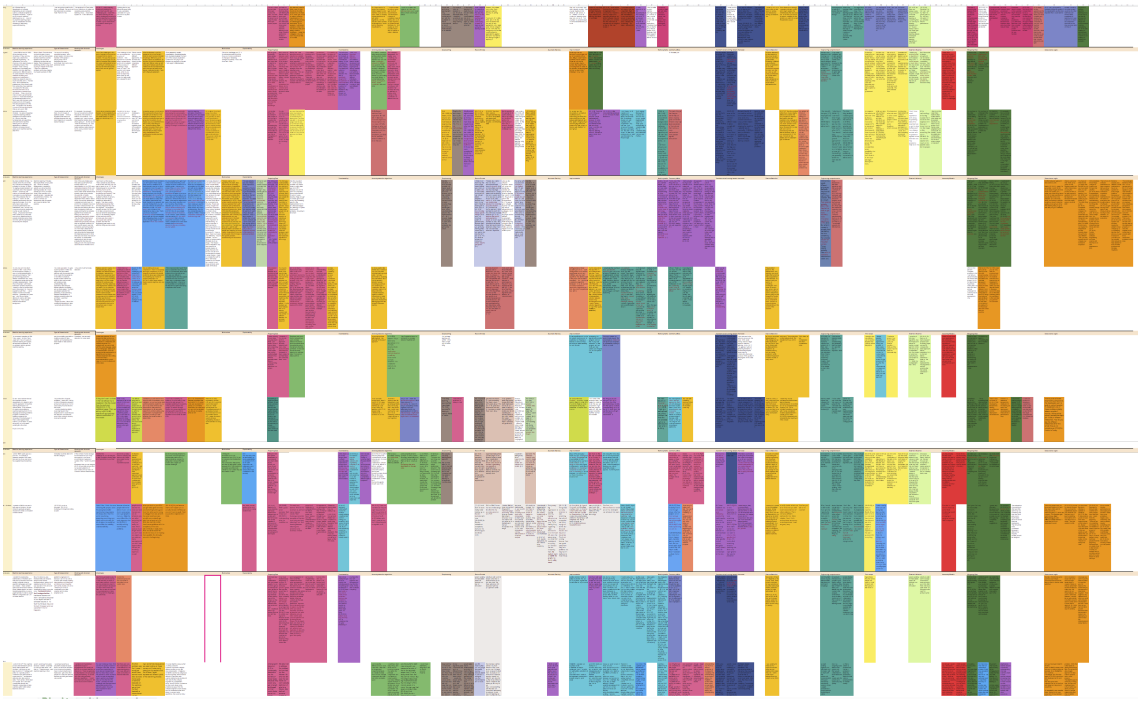
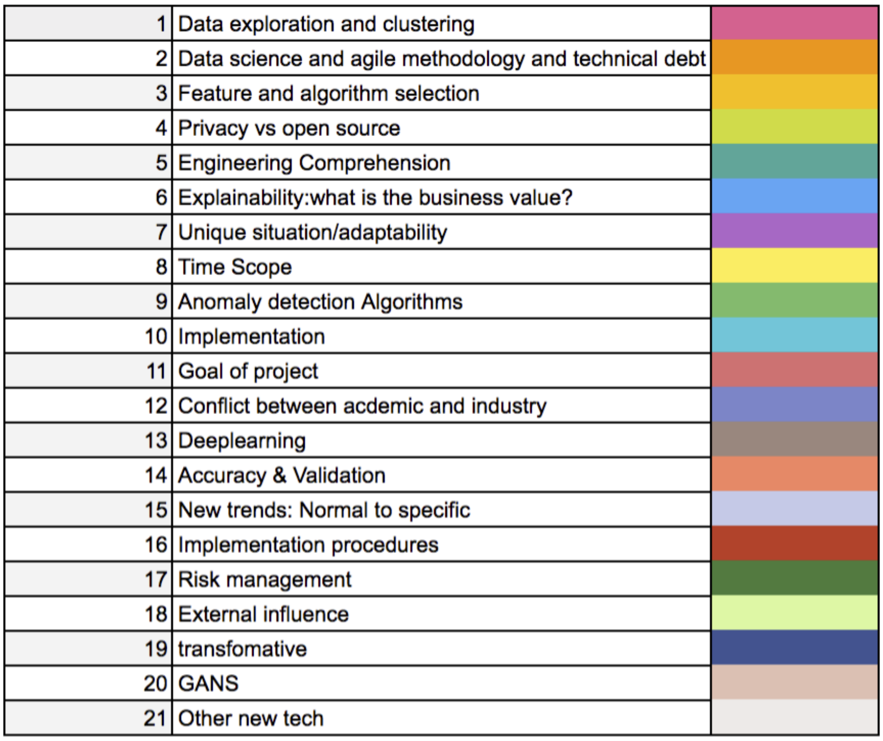


Figure 3.6: Part of color labeling table used to distinguish different themes.



1	Data exploration and clustering	🟥
2	Data science and agile methodology and technical debt	🟧
3	Feature and algorithm selection	🟨
4	Privacy vs open source	🟨
5	Engineering Comprehension	🟩
6	Explainability:what is the business value?	🟦
7	Unique situation/adaptability	🟪
8	Time Scope	🟨
9	Anomaly detection Algorithms	🟩
10	Implementation	🟦
11	Goal of project	🟥
12	Conflict between acdemic and industry	🟦
13	Deeplearning	🟫
14	Accuracy & Validation	🟧
15	New trends: Normal to specific	🟪
16	Implementation procedures	🟥
17	Risk management	🟩
18	External influence	🟩
19	transfomative	🟦
20	GANS	🟫
21	Other new tech	⬜

Figure 3.7: Theme headlines with their color in color labeling

3.3.4 Interaction analysis

Looking at the visual representation of the color labels it became evident that some colors where reoccurring more frequently than others. Runeson et al. (2012) also suggest to draw a structure to the themes in hierarchical order and to give them priority in the analysis based on the times the themes were mentioned. This was done at the beginning of the chapter 4.

Some of our most occurring themes where: the importance of preparing data, the explainability of models and implementation procedures. We will present the different findings in ascending order. This observation will then reflected in chapter 4 and chapter 5 by presented.

3.4 Practical verification

As a supplement to the qualitative case study, a practical experimental verification of the case study's result was carried out. The practical verification included implementation of IBM's anomaly detection system and running the system on new data gathered from a prospected client, namely Energy Machines. The purpose of the experiment was to investigate and confirm the findings in chapter 4 and to test the feasibility of reducing the maintenance cost for the client company by alarming for leakage before it occurs.

The process overview of practical verification is shown in Figure 3.8, the two overlapping parts in the diagram and the final results will be described in section 4.6. In the following sections, the first step of preparation in terms of the client and systems will be explained.

3.4.1 IBM's cognitive anomaly detection system

IBM has aimed at developing an ML cognitive anomaly detection system(CADS) that is adaptable towards new customers. The CADS takes real-time data from industry machines' sensors as input to two ML models (Boltzmann Machine, Isolation Forest) and then outputs anomalies in the data flow which could potentially indicate inference. As shown in Figure 3.9, the CADS system shows the status of the selected sensors: for example, the dark blue line in the chart represents the real data from the sensor tagged as SITE.ASSET71KL630 in cluster CONVEYOR_CLUSTER_16 by using model 1, the light blue line represents the prediction value which apparently has similar value with the dark blue line, and the anomalies points are marked as red points in the diagram. The vision is that this type of ML predictions could lower scheduled maintenance since the tool alerts an operator before a small anomaly

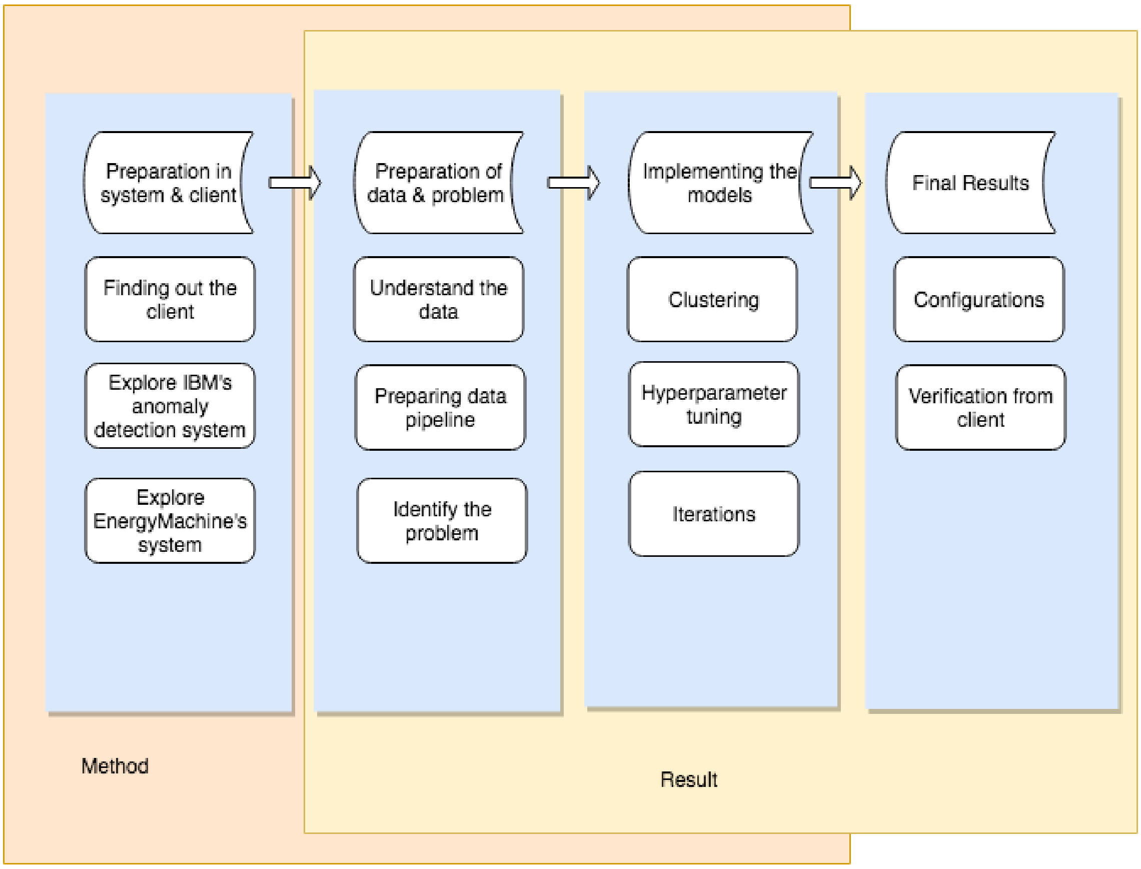
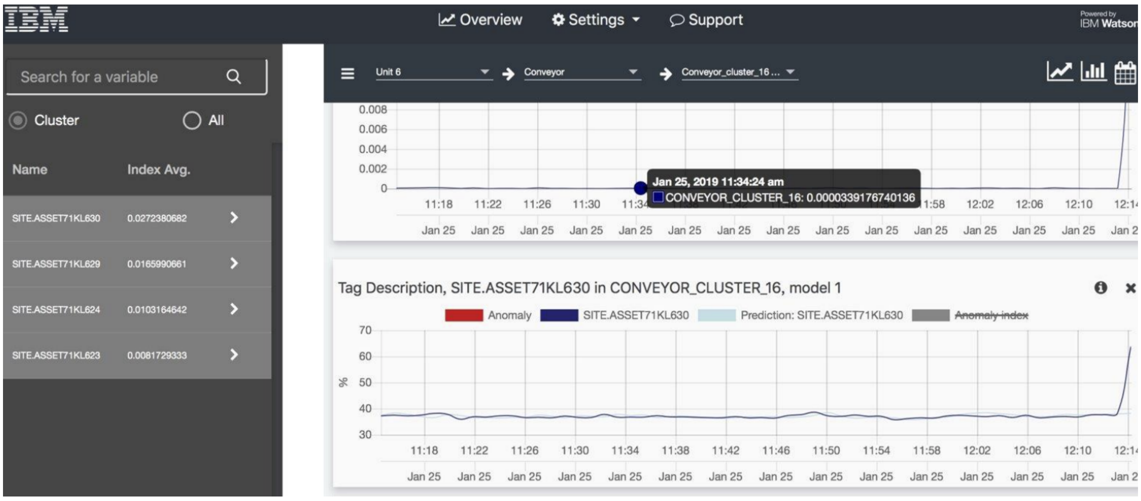


Figure 3.8: Overview of method and result for practical verification

grows into a potential failure.



Name	Index Avg.
SITE.ASSET71KL630	0.0272380682
SITE.ASSET71KL629	0.0165990661
SITE.ASSET71KL624	0.0103164642
SITE.ASSET71KL623	0.0081729333

Jan 25, 2019 11:34:24 am
CONVEYOR_CLUSTER_16: 0.0000339176740136

Tag Description, SITE.ASSET71KL630 in CONVEYOR_CLUSTER_16, model 1

Figure 3.9: Example view of IBM's Cognitive Anomaly Detection System

3.4.2 Finding out the clients and their system

In parallel with the case study preparation described in section 3.1, the search for a suitable client with corresponding sensor data commenced. The researchers contacted companies such as Renova, Göteborgs Energi, Sweco, Energy Machines and Västtrafik, requesting their data for future testing and model training. After communication through emails, presentation, and meetings, the final chosen client was Energy Machines. The reason to chose them was that Energy Machines reacted actively for the exploration and expectation of new ML technology during the discussion, as well as they had their own existing human-machine interface(HMI) named ControlMachines(TM) for monitoring their sensor data. Energy Machines as a company offers two products: ClimateMachines(TM) the air handling unit and EnergyMachines(TM) the heat pump and delivers it in complete designs on integrated energy systems on cooling and heating. The HMI of Energy Machines lets users monitor the status plants and sensors (see Figure 3.10), it also records operations and alarms in the plant.

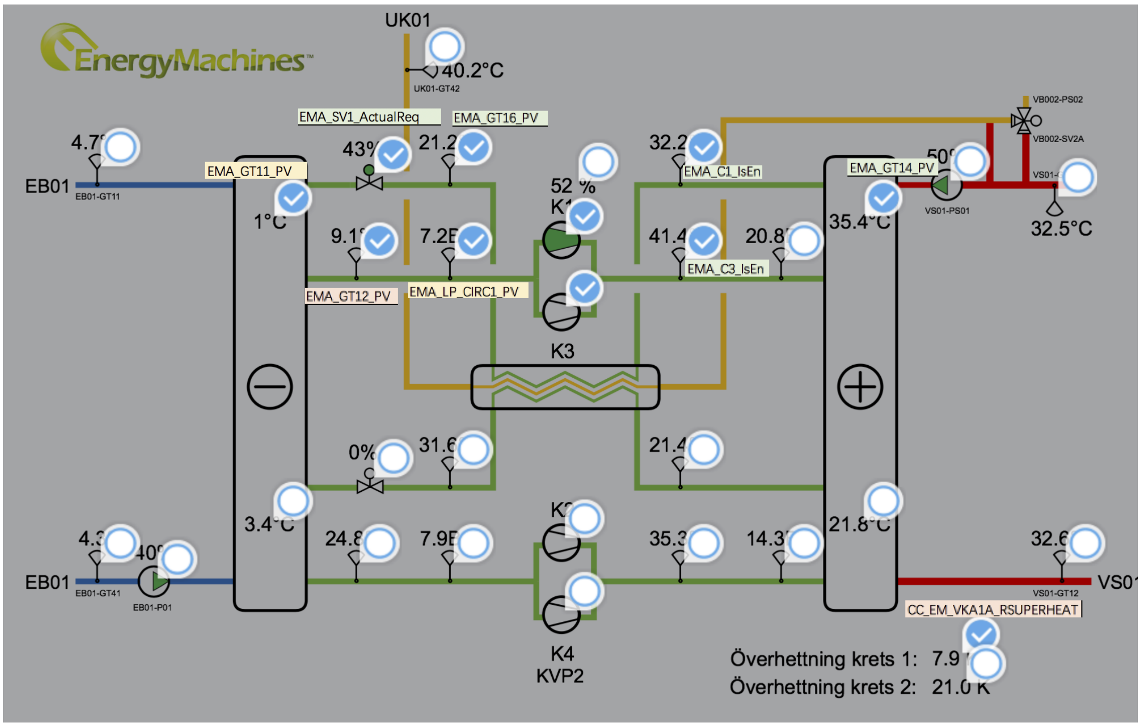


Figure 3.11: The structure of sensors related to main leakage problem

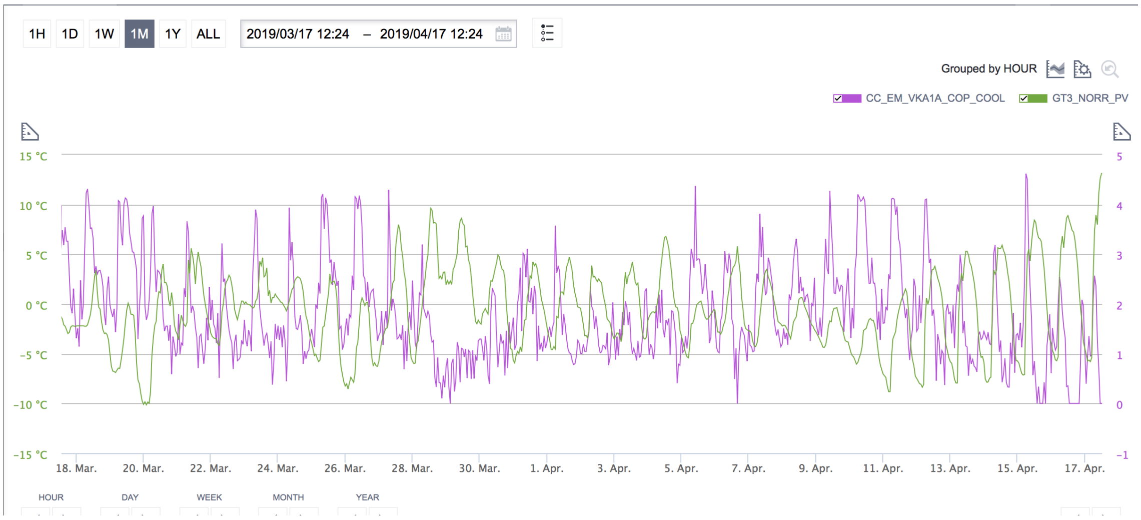


Figure 3.10: Trend of sensors in ControlMachines(TM) HMI

3.4.3 Identify the problem

During the meetings with Energy Machines, they stressed that their highest priority was to identify when there had been a leakage in the refrigerant system. To help the researchers in their search they provided seven sensors out of around 60 which they thought would show the highest correlation with a leakage. These sensors would then be used for clustering (see subsection 4.6.2). However, after continuous dialog and several iterations of tests, the final number of identified sensors with possible relation to the leakage problem was 11, see Figure 3.11.

In order for the researcher to distinguish a regular outlier from an anomaly, Energy Machines explained their hypothesis of the leakage phenomenon; First, the expansion valve E of sensor EMA_SV1_ActualReq is going towards 100%, where the rate of increase speed dependent on the size of leakage. In the next situation, there is a value X of the subcooling system is calculated by condensing temperature minus liquid line temperature(sensor value EMA_GT14_PV - EMA_GT16_PV in Figure 3.11). The key phenomenon is that the value X starts to go toward 0 kelvin or negative 2 to 3 Celsius. More specifically, one degree's change within 30 minutes and the expansion value deviate from normal operation. So this means that the abnormal operation is when the E value is open more than the normal state while the condensation is low. After the value E up to 100% open, the suction pressure(EMA_GP11_PV) and evaporating temperature(EMA_GT11_PV) will decrease. All these situations leading to decreasing COP of the unit efficiency or break down, and superheat will be increased at the same time. As a result, we put all related sensors into the physical cluster as shown in Figure 3.11. The interactions inside those sensors would be further analyzed after inner clustering described in section 4.6.

4
Results

Our result contains two parts, the main part is organized from the qualitative case study, which were divided into five main areas: techniques, data scientist, software engineer, management and client. The themes are divided based on different roles in ML projects, where Techniques is also extracted as a separate area since it is used by other roles, their relations are shown in Figure 4.1[1]. The detailed hierarchy of sections and themes is displayed in Table 4.1, where all the identified themes are presented with a short explanation with the corresponding mappings to research questions. Many themes are interconnected and can not be strictly separated, hence, there may be reoccurring statements in different parts of the result section. The second part from section 4.6 is the result from our practical verification test with IBM's anomaly detection tool and Energy Machines Data, whose method also was described in section 3.4.

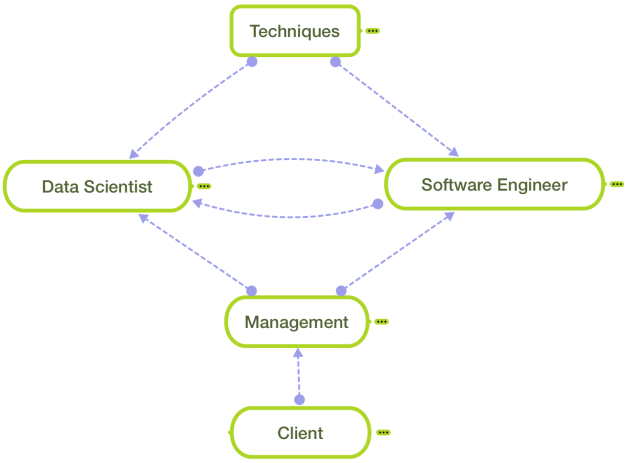


Figure 4.1: High-level mind map of roles and their interconnections identified in the study.

[1] Areas fell into 4 high-level roles and 1 technique, which are extracted from the list of questions, see Appendix A and the resulting themes are extracted from the transcripts, see subsection 3.3.2 and color labeling in subsection 3.3.3

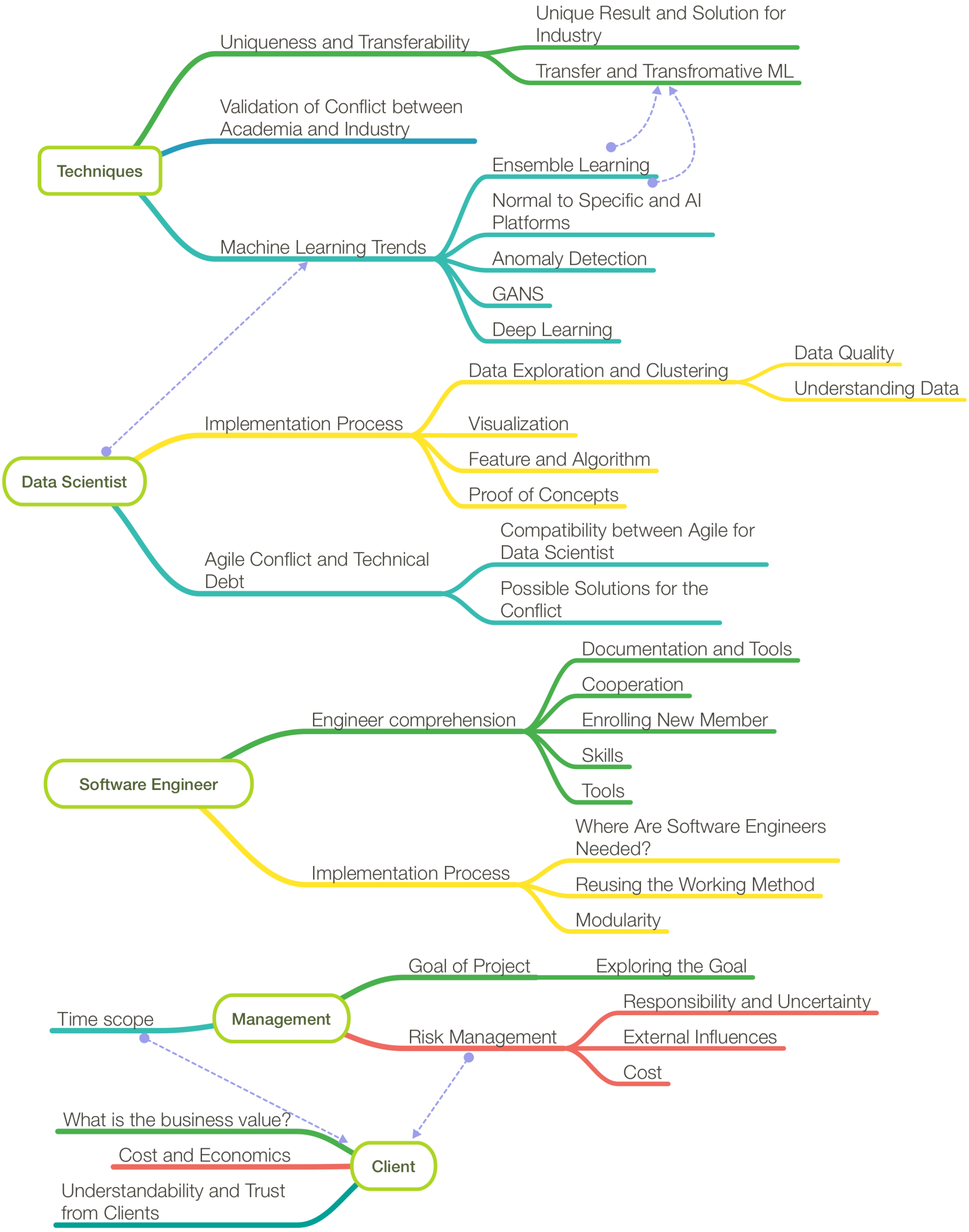

Techniques
Uniqueness and Transferability
Unique Result and Solution for Industry
Transfer and Transfromative ML
Validation of Conflict between Academia and Industry
Machine Learning Trends
Ensemble Learning
Normal to Specific and AI Platforms
Anomaly Detection
GANS
Deep Learning
Data Scientist
Implementation Process
Data Exploration and Clustering
Data Quality
Understanding Data
Visualization
Feature and Algorithm
Proof of Concepts
Agile Conflict and Technical Debt
Compatibility between Agile for Data Scientist
Possible Solutions for the Conflict
Software Engineer
Engineer comprehension
Documentation and Tools
Cooperation
Enrolling New Member
Skills
Tools
Implementation Process
Where Are Software Engineers Needed?
Reusing the Working Method
Modularity
Management
Time scope
Goal of Project
Exploring the Goal
Risk Management
Responsibility and Uncertainty
External Influences
Cost
Client
What is the business value?
Cost and Economics
Understandability and Trust from Clients

Figure 4.2: Overview of roles with corresponding themes.

ID	Theme	Explanation	Research question
1	Data exploration and clustering	Data understanding, cleaning, developing data pipeline, etc	RQ1
2	Data science and agile methodology and technical debt	Is there any conflict for data scientist working in Agile method?	RQ2
3	Feature and algorithm selection	What features and which algorithm should be used for a better result?	RQ1
4	Privacy vs open source	Data, algorithm privacy, model privacy, possibility to build upon other researchers' existing models	RQ2
5	Engineering Comprehension	Collaboration and ability to understand, how to fit in the new group	RQ2
6	Explainability: what is the business value?	How to explain the prediction result and how the model works, what are the challenges towards customer qualification	RQ2
7	Unique situation/adaptability	Possibility to reuse the working method, algorithm, data, model, system, etc	RQ2
8	Time Scope	Measure the scope of the project in terms of time and money	RQ3
9	Anomaly detection algorithms	This is preparation for IBM anomaly detection system and our practical verification part	RQ2
10	Implementation	The process for developing a Machine Learning system.	RQ1
11	Goal of project	Project goal in research and engineer perspective.	RQ1
12	Conflict between academic and industry	Time, accuracy, project scale, the customer.	RQ2
13	Deep learning	DL is a highly discussed trend.	RQ3
14	Accuracy & Validation	Validate the ML result, improve the accuracy of the model.	RQ2
15	New trends: Normal to specific	AI to AI, broad AI to specific AI	RQ2
16	Implementation procedures	What are the process and challenges in implementing the model to the system, to make ML into the industry.	RQ2
17	Risk management	Could the model be trained or retrained automatically?	RQ3
18	External Influence	If new technology new algorithm comes, what to change	RQ3

19	Transformative	retrain the model or set up a new model based on an existing model	RQ3
20	GANS	Generative Adversarial Networks are deep neural net architectures comprised of two nets, pitting one against the other	RQ1
21	Other new technology in ML	The trends of ML in interviewer's perspective.	RQ3

Table 4.1: Identified themes and corresponding explanations.

4.1 Techniques and conflict

In this section, the most reoccurring technical challenges and that interviewees face when applying ML are presented together with corresponding solutions used by the interviewees. The section also covers future trends and technologies related to possible future solutions.

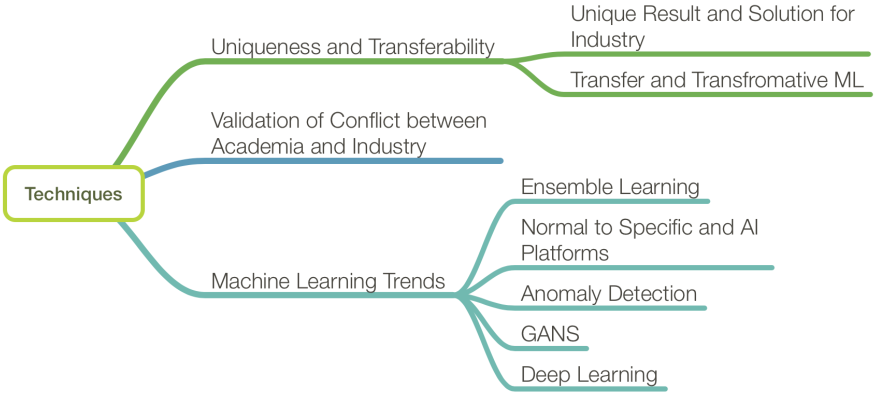

Techniques
Uniqueness and Transferability
Unique Result and Solution for Industry
Transfer and Transfromative ML
Validation of Conflict between Academia and Industry
Machine Learning Trends
Ensemble Learning
Normal to Specific and AI Platforms
Anomaly Detection
GANS
Deep Learning

Figure 4.3: Result Mind Map for techniques

4.1.1 Uniqueness and transferability

All interviewees talk about the uniqueness of projects. The challenges of transferring and reusing any previous results to new projects are hard since customers have different data, goals, and environments. However, there are some techniques such as AI-platforms which are up and coming that may alter this in the future.

4.1.1.1 Unique result and solution for industry

One challenge of applying ML in unique industrial settings is related to extracting the right mathematical expression for the industrial problem. The reason behind the difficulty of extracting mathematical expressions is that, although the theory behind ML has existed for a long time, it is only now that the computational power is available to the general public, and also for more and more available universal education for ML. There can also be restrictive requirements that limit the project, such as it may be prohibited to use multiple customers' training data in the same model. Still, it is the feature extraction and problem formulation that sets the overall restraints of the industrial adaption of ML according to the interviewees.

> "Principles of machine learning and AI are simple but we cannot apply these principles to any situation. The problem is that we have many many different situations in the real world so one specific real-world application is different than others." - participant 8

Another interviewee states that creating a custom piece of technology (through a pilot study) and making it applicable in other areas is typically not possible, which is also the reason why it is usually very expensive to conduct ML projects. However, the person in charge of the project should be aware of the resources required to scale and also the learning outcome of the project. The big question for practitioners is how much progress, in terms of learning outcome has been made.

> Will we be able to reproduce this (ML project) faster and more efficiently next time? - participant 6

4.1.1.2 Transfer and Transformative ML

Transfer ML is the act of reusing pre-trained models designed for a specific problem as a starting point for another model. Furthermore, transformative ML is a specific transfer method and different interviewees seem to have different views of what the concept really means. However, there is a high-level agreement that it specifies reusing data and already trained models, e.g. by reusing features from vehicle recognition in car recognition. Transfer and transformative ML is mentioned by nine interviewees, however only one had a positive experience with reusing pre-trained models in other projects and domains, others did not have experience for reusing and most of them think reusing models especially in a specific industry scenario is unrealistic.

> "Both, we are going from storage to server so that's the same field if you will. But we are also trying to now move for example to wind turbines and wind turbines are a completely different type of equipment." - participant 10

The overall consensus was that transfer model could work in theory but that it is not used in practice due to the difference in data and industrial uniqueness that described in subsubsection 4.1.1.1. Many interviewees have tried it without succeeding.

> "You can not just simply take something from the shelf and apply it to your specific data, you need to know or understand what's going on." - participant 1

However, features from similar data can sometimes be reused and the field is being explored by some of the bigger IT-giants. IBM and Google are for example developing platform services that are meant to cut out the data scientists work, which is discussed in subsubsection 4.1.3.3. It is becoming more and more apparent that companies want to reuse previous investments, especially in images recognition, natural language processing, and customer identification. This is elaborated in subsection 4.1.3. In deep learning, for example, transfer ML could mean retraining the last layer of the neural network as mentioned in the second quote below. Related results are reusing the working method and increasing the modularity of the system, which will be described in subsubsection 4.3.2.2 and subsubsection 4.3.2.3.

> "You can take a part of that model and add a small extra part of it which is specific to your problem. Then you retrain it but still rely on what was done before. So then if you had your own image recognition problem then you don't need as much data for that problem. This is typically a cheap way to get good accuracy first...retraining or adapting the model to a specific task is one way of reusing previous work, in deep learning that could mean the last layer of the learning network..." - participant 9

4.1.2 The gap between academia and industry

Seven interviewees also mentioned the clash between researches at university and industry. The bigger IT-giants have the advantage of huge computational power and data pipelines that make for the worlds most promising science in the ML field. Companies see value in publishing their research in order to attract attention and raising their stocks. However, it is almost impossible for the average researchers to replicate these studies since they do not have access to the surrounding support structure needed, such as computational power and data streams. This leaves a gap between top research and applicable ML since it is not accessible to academic units and as a result, less validated. One interviewee mentioned that the IT-giants might need to start providing their support structure for the researchers who want to replicate and improve IT-giants' studies as a possible solution. Other interviewees mention that some companies already provide this but that it is complicated to get them through the process.

> "Today you get the data, you get the software, everything, for example, you just don't have the computing power, right I mean who has even tensor units to train these humongous deep-nets, it is apparently because of its size. So the top research in machine learning these days is not reproducible by academic units" - participant 1

Another clash is the gap between research and industry, the scientists who conducted the most innovative academic ML researches have a hard time translating it into something that can generate value in an industrial context. Universities often have a research solution looking for a problem while the industry has a problem looking for a solution.

Another restriction is that in some cases researchers do not share their code with their publications. If the access to the code is not given by the author, it will be harder for new researchers to reconstruct the experiment through the description of the article, industry included, to replicate and ultimately make use of it. One reason for this might be that some the author works under a time constraint and the code is too dirty to share.

The interviewees also expressed that the industry has been naive about the time scope to integrate ML. Everybody has started to gather data in large data lakes but few actually have any idea about what to make of it. One interviewee expressed that this will probably be solved in ten years when data science education and computational power is more widely accessible.

4.1.3 ML trends

Ml has been popular in recent years as an effect of more accessible computational power. The theory has existed since the eighties, but it is not until recently that it is has become usable on a broader scale. However, most interviewees agree that ML and AI may have been overestimated in terms of how quick the industry would be able to make use of the technology. What is certain though, is that industry and educational institutions are catching up and that in the coming years the clearer effects and use cases will be seen. In the following, we introduce the trends discussed by the interviewees.

4.1.3.1 Anomaly detection algorithms

Since the thesis has a practical focus on anomaly detection, one specific question in the interview guide targeted if the interviewee knew of any particular useful algorithms or use cases in anomaly detection. Some interviewees had no experience of anomaly detection but in total nine people answered the question related specifically to this subgenre of ML.

One challenge with anomaly detection is that there are generally too few failures to do any properly supervised learning. Which makes it difficult to train and verify the models on historical data sets. Hence, data scientist usually resort to unsupervised ML. One possible solution though as mentioned by one interviewee is to create fictional anomalies (manually made anomalies). But then these anomalies need to be validated in the context of the real anomalies, which adds the complexity to the project.

Another problem with anomaly detection is that there are no specific anomaly detection algorithms as the quote below suggests. The traditional method to tackle this is to try a bunch of different classification techniques that distinguish outliers. These outliers might not even be anomalies if they are expected or allowed by the operator. One possible technique mentioned was XGboost which is a ML library specialized in gradient boosting which is used for regression and classification problems. Another reoccurring idea was to use high dimensional reduction and clustering in order to see multivariate movements in the data.

> "There is no special anomaly detection algorithm, you need to try a bunch." - participant 10

The majority of the interviewees also express that a simpler model such as linear models and tree models are preferable in most cases for anomaly detection since they work on smaller data sets. The tree models like random forest are also more forgiving in terms of data completeness and quality. Another technique mentioned is Gated Recurrent Neural Networks (GRU) over Long Term Short-term Networks (LSTM) but that these complex models are more dependent on the use case of which they are supposed to fit.

4.1.3.2 Deep Learning

Ten interviewees talked about deep learning (DL). DL is a large neural network and has been around since the 70s, but only now we have the computational power for DL as previously stated. One interviewee said DL can find more fine-tuned patterns that humans can not see such as in face recognition. Reinforcement learning like

AlphaGo[2] has mostly been successful in academia and industry. One interviewee talked about a recently being researched technique in DL is adversarial machine learning, which attempts to fool models by malicious input.

DL need huge amounts of data. Hence, DL is somewhat restricted to various fields of vision, speech and video game research, since these areas have a lot of dense information in them. The data requirement also restricts the applied field of deep learning, some interviewees mentioned that clients express that they want the data scientist to try deep learning or other complex methods which in most cases are unrealistic since the amount of data in the clients' companies are not enough. For example, one participant described that their client had 20 years worth of finical stock market data. This was still not enough for successfully implementing one of their complex models.

However, there are areas as the ones mentioned above that naturally have dense information in them. There are also some public records available, for example, one interviewee mentioned that if a researcher signs the right forms one could get access to huge data sets of medical records to be used for cancer prediction or other more general approaches.

> "Complex models are not applicable to small amounts of data. Big data is limited to the web domain. Small data could be for example 20 years data in the financial industry." - participant 16

In addition, computational power and time are other restrictions for deep learning. The following cost of deep learning is higher both from computer power, time and human part. Higher computational cost may give access to high performance for better accuracy, but in the real case, the higher accuracy approach could not deliver on the trying. So use the lower cost approach for backup is a better solution. Deep learning is nice to try but not suitable for every case.

> "So we had another person doing a deep learning LSTM model. But that took several months for him to do that. And he was training and training and training and it was taking forever, in the end, it turned out to be good." - participant 14

4.1.3.3 Normal AI to specific AI and AI platforms

Another reoccurring area from the interviews is the development of AI-platforms. Some of the larger IT-giants see value in capturing previous research and embedding

[2] AlphaGo uses a tree search algorithm to find its moves based on knowledge previously "learned" by a deep learning method.

it into reusable formats. Since the number of ML projects is limited by the number of people who can drill into the data. In short, they aim to build platforms where ML modules and code can be reused in an accessible way for third parties, as already mentioned in section subsubsection 4.1.1.2. The idea is that people will be able to buy and use different kinds of advanced analytics without extensive coding. Normal software engineers can easily apply the methods on the platform. AI infrastructures are boiling down the science to practice.

Most of the businesses who invest heavily in ML and data mining today are the industries that are cash rich and data-rich, meaning it is mainly the IT-giants, financial industry and the insurance industry who implements the technology. However, with AI-platforms more industries can potentially capture value from ML. Participant 1 forecast that this together with scientific progress will lead to more disciplines becoming increasingly data-driven. And as a result that the data science field will change to be more area specific since a single data scientist cannot cover everything. Just like Bio-informatics have transformed during the last 15 years to involve specialized data scientists.

> "For example, Microsoft's have their own ML ways that we can drag and drop their models. To make it more available for people. Host your Microsoft service model and let the developers use whatever they want." - participant 12

ML works best on well-defined models, so the result crucially depends on how the problem is described mathematically. Data scientists can spend a significant proportion of their time making a problem more explicit in order to make use of the technology: *Is it computational expensive or not? Where is the main bottleneck?* Utilization of techniques and platforms is one way to make the process repeatable. For example, a methodology called CRISP-DM (Cross Industry Process for Data Mining) can be used in data analytic projects for business understanding said by one interviewee.

> "It's the expertise that's the bottleneck, at least form an economic perspective. If you can capture the experience in an asset that allows regular people to exploit the resources you can lower entry costs. You can have a much larger impact if less detailed skills can use the encapsulated experience. Making it repeatable is key." - participant 6

One senior interviewee also talked about the future trend of normal AI and specific AI, which means from ANI to BAI to make the AI more human-like and adaptable for various problems. This is described in subsection 2.4.2.

> "When there are more common methods/validations it will eventually automate or at least be more usable. " - participant 9

4.1.3.4 GANS

Two interviewees specifically mentioned Generative Adversarial Networks (GANS) as the new hype. GANS are deep neural net architectures comprised of two nets, putting one against the other. GANS models can be trained to make up high-quality images of fake human faces that look almost completely indistinguishable from ordinary faces. Impressive technology that still is in the bud but that in the future perhaps could be used in data augmentation for video simulations for self-driving cars. Another idea that is being explored is using GANS for balancing out labeled data in supervised ML. In short, creating own samples through GANS that could help to balance the minority vs the majority labels.

4.1.3.5 Ensemble learning

Six participants talked about ensemble learning as an interesting future area but only three had actually tried it out in practice.

> "I think several times you find it hard to select just one model and then you're probably better to combine several models with some logic and more kind of intelligence, putting in that voting system, for instance. So I think we will see more of that in the future." - participant 14

> "We think actually assembling models would be very useful in another project about wind turbine, because the idea around them is, you may actually find that some models work best in certain conditions or for certain subproblems of your problem and instead of trying to find that model that does it all, which may not always be possible. It may be worth investing in building three to five models and letting them all running parallel each for different scenarios and then combined the results." - participant 10

Ensemble learning is a combination of multiple learning models, where the structure among these models can be flexible. As previously mentioned, multiple models and tests can increase the validity of predictions. Ensemble learning is one technique where the output of other models is used as input for an ensemble model, i.e. a multilevel prediction. Since Narrow AI are optimal for certain sub-problems one use case could be to let different models run in parallel for different scenarios as expressed by the quote below.

> "Some models work best in certain conditions or for certain subproblems of your problem which is not possible to find that model. Building 3, 4, 5 models and letting them all running parallel each for different scenarios and then combined

> the results could get better results than just one fits all." - participant 10

4.2 Data scientist

This section covers themes related to the role of a data scientist and strategies that can serve as potential support in ML projects.

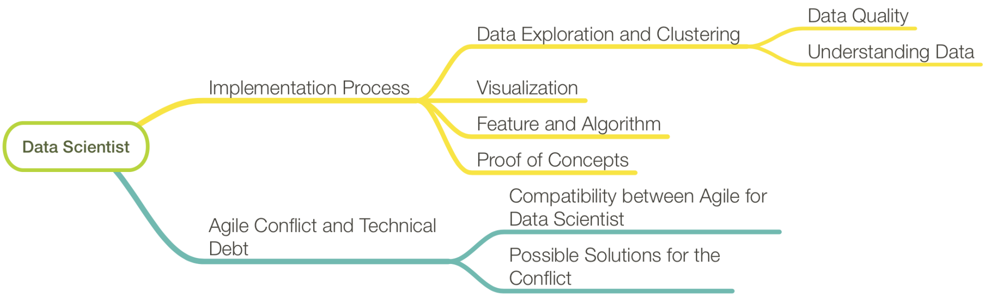

Data Scientist
Implementation Process
Data Exploration and Clustering
Data Quality
Understanding Data
Visualization
Feature and Algorithm
Proof of Concepts
Agile Conflict and Technical Debt
Compatibility between Agile for Data Scientist
Possible Solutions for the Conflict

Figure 4.4: Result Mind Map for data scientist

4.2.1 Implementation process

The ML implementation process consists of different data science activities that together create a usable ML model. In this section, some of these activities and structures are explored.

4.2.1.1 Proof of concepts

One task in the implementation process is mitigating risk through small scale tests such as Proof of concept (POC), pilot studies or experimental phase. The majority of the interviewees expressed that when working in industry one rarely launch a large scale project right away. There is usually a POC that exist for trying out the uncertainties, delivering small but quick value for the clients. After that, a Minimum Valuable Product(MVP) could be created for future collaboration. The challenges and technical debt that may arise in POC are usually related to the scale of data. The models created during a POC might not work for larger data sets, or the real data is different compared to the data provided in the POC.

> "Where you're deploying the models, while in real case you have a lot of more data, but when you do it in your POC you get really good accuracy and everything's pretty good. And then when you're trying on a bigger than data set with the same model, the accuracy actually very suddenly didn't get as good as in the POC because there are some problems with the complexity of the data, these type of issues that we haven't thought of." - participant 11

Also, important misconception from clients is that the POC can be transformed into a large scale without much additional work, while in many cases, it may need to be rebuilt from scratch. To tackle this the team can aim for higher modularity that makes it easy to switch from one component to another, which is described in subsubsection 4.3.2.3.

> "I really think POC should be small. We should not focus on making them scalable. But let's play around (with the data) in the beginning. Clients don't expect you to actually recode stuff because you don't want to scale (in the beginning)." - participant 12

4.2.1.2 Visualization

Five interviewees talked about the use of visualization during the preparation phase of data as well as in observing the prediction result.

Visualizations such as a graph of the distribution of data can help data scientist get an overview of the data. It can also be used when talking to domain experts when looking for explanations of bad quality data such as null values and abnormal values. Visualization at the beginning of projects also helps data scientists to understand the structure of the data, correlations and potential trends, for example, by making a histogram of different features. It can also be used in order to validate the prediction result, for example in the anomaly detection area. Data scientists can visualize the anomaly index (model output) with actual sensor values, presenting the anomaly scenario to the customer and validate the predicted result with the domain expertise. Another example mentioned by the interviewees is to compare the model's anomalies with anomalies that observed by industry system operators: is it a clear disruption from the normal data distribution?

> "For cleaning up the data, we need to look at the whole debate... I also want to do visualize and see if the detection has any fault and the visualization is always right to do if you have time." -participant 4.

4.2.1.3 Data exploration

As previously stated a lot in ML boils down to the data. All interviewees agree that even if a project involves the most promising researchers and engineers, it is still hard to get any useful results if the input data is off. Or as neatly described by one of the interviewees: "garbage in, garbage out". Hence, one important aspect in embedding a ML algorithm in an existing software system is to first explore and clean the corresponding data. The theme seems to suggest not to start with the cleaning right away, instead, the interviewees usually set aside a few days for data exploration in order to get some insights on a general level. If the data is not good enough, the common practice seems to involve writing a recommendation of required fixes for the client before taking on the project. Furthermore, some researcher express frustration over that people assume that once a data scientist has the data they can solve anything, but in reality, that is not really the case.

> "They (clients) just throw us the data and says: look at it and maybe you can find something" - participant 7

4.2.1.4 Data quality

The initial gathering, screening, exploration of data takes the majority of data scientists time and energy. Some of the necessary aspects that need to be taken into consideration in the initial phase are the amount of data available, seasonality of the data, unit of the data and the volume of the data. This can be done through various visualization methods such as histograms and distribution of data points as described in subsubsection 4.2.1.2. One interviewee expressed that in industry, data rarely comes neatly in a CSV format, which is common in university projects. As a result, the data scientist usually writes their own parsing and create their own data flow since raw data is common.

What a data scientist are looking for are data corruptions of different characters such as missing values, outliers or features that do not have any predictive power. When these types of data points are identified the data scientist starts the filtering phase which involves removing all the above mentioned noisy data points. Some interviewees stressed that cleaning data semi-manually is usually hard and time-consuming and quickly becomes unfeasible in large scales. It is an expertise that requires extensive experience which in some cases results in a bottleneck in developer teams since no one else can help in the decision-making process.

> Consider if it's worth doing imputation. And so many different methods of doing imputation. Also if you see missing values you can try to figure out the root cause." - participant 12

4.2.1.5 Understand the data

Eight interviewees mentioned that in order to understand the data it is also necessary to plot the data, consider data normalization and data scaling, what are the minimum and maximum values. Furthermore, they emphasized the importance of understanding the architecture of the machine and sensors that are being targeted. For example, if there is no fixed max value for a data point one, the option is to calculate average max value over a period of time and assume a maximum value. One interviewee also pointed out that a data scientist can learn a lot from missing classified data, false positives: what are the correlations between the missing classified data points. It is also a question of balancing potential value and ethical aspects: is there a difference between a financial instrument and a cancer patient in terms of miss-classification?

> ...if my algorithm predicts that this guy doesn't have cancer but he or she does then the cost is very high... - participant 5

To effectively get to the root cause of missing values, some interviewees mentioned that discussion between the client and developers are needed. In other words, the team needs to involve the subject matter expert. This can be done through a workshop with the people who got the knowledge, gather all information that has a direct or indirect impact. However, asking domain experts may also introduce bias as formulated by one interviewee.

> "The downside of asking customers or close collaboration with domain experts is that might be biased, but it's kind of been working quite well." - participant 13

4.2.1.6 Feature selection

Feature selection and clustering may be tricky since it depends on the use case:

> "For example clustering in documents, come up with a division into groups. So what is the right division into groups depends on what we want to use it for." - participant 9

One reoccurring theme is the importance of gathering domain knowledge in order to select good features. Data scientists require insight into the underlying mechanisms of that field to interpret results properly. Having a proper business understanding can help in this process. The feature extraction can be done either through the researchers own experience or in combination with a feature selection algorithm.

Also, it is unwise to not use the extracted features as it might result in loss of important patterns as expressed by bellow quotation. However, a model might also be strengthened by leaving out redundant features.

> "You can't just drop features because you can't interpret them. You have to replace it with something meaningful, if we just drop we have empty space there, we might lose this very very important pattern." - participant 8

One interviewee talked about an end to end machine learning where the model can pick out the features itself automatically rather than manually look at the weight of the variables in the linear model.

Another approach is to separate feature engineering from the actual model. For new problems a data scientist may use a separate feature extraction approach, starting with a pre-trained model that has pre-learned features which can be reused. Otherwise, the features are dependant on the domain knowledge shared by the client.

4.2.1.7 Algorithm selection

It is also a reoccurring statement among all interviewees that it is hard to choose which algorithm to use. A data scientist is required to understand the details of multiple algorithms in order to pick the optimal one. This is challenging since the number of available engineering choices can be exponential in the search phase. Furthermore, the interviewees agree that complex models have a tendency to overfit more easily than simpler models since they usually require a larger quantity of quality data. Instead, they suggest to start with something simple and if successful, extend with more complex algorithms. For example, subset selection, random forest or Lasso may be used to do some initial analysis depending on the problem. Many interviewees seem to argue for easier algorithms simply because those are usually the ones that make it into production.

4.2.1.8 Skills and other challenges for data scientists

Similar to software engineers described in subsubsection 4.3.1.2, requirement regarding skills is also a challenge for a data scientist. Especially younger data scientists may have a hard time making a good connection between high-level goal descriptions of a task and the mathematical formulation of the task. Hence, sometimes there needs to be an experienced data scientist that can help in this process. For selecting the algorithms in ML, all interviewees agree that data scientists should have a background in understanding the mathematical aspect of the algorithm, not just use it but also understand why it works. This helps in the selection process in choosing the right algorithm and also reduce the risk of needing to change later on.

> "Like I have interest in what data scientist said during the meeting, so I pick up the literature and I read and I noticed the data scientist they follow up so much on their progress of research." - participant 2

Another reoccurring theme is that data scientists are becoming more specialized, mainly because it is unreasonable to cover all topics, like anomaly detection, text analysis, and NLP. However, programming languages used by data scientists are limited to mainly Python, R or SPSS. Since there is no magic tool for all, sometimes data scientists have to know all these languages with corresponding libraries, which is difficult and takes time.

4.2.2 Agile methodology and technical debt

In this section, we present the theme of agile methodology that raised from the interviews. Some interviewees expressed negative experiences with following agile methodology and the technical debt that may arise as a consequence.

4.2.2.1 Compatibility between agile and data scientist

Three of the more research-oriented interviewees mentioned that agile methodology can be tricky to implement in data science projects. Agile methodology focuses on quick value bringing activities and stresses the importance of short iterations. This seems to be in conflict with the research aspect of data science that aims for high performance through a rigorous exploration of different methods and models. Clients in many cases do not care about the details of the model, they only care about whether it works or not which might cause a misalignment between researcher goal and client goal. However, ten interviewees expressed that agile methodologies worked just fine in data science and some even stated it is a key aspect in the data scientist consultant role.

When working with Agile methodology it is essential to have a continuous dialogue with the customer in order to validate progress. According to the ten interviewees, this methodology very much aligns with the data science-consultant role, especially in POCs where the time scope and resources are usually quite limited.

However, the research-oriented interviewees also expressed that code developed too quickly might cause problems later since it may be hard to remember exactly how the code is structured, in short, technical debt may arise. And as a result, the prediction might not prove to be as good on a larger data set when the surrounding infrastructure is limited.

> "...if you have more research-heavy projects then, of course, it's possible to questions the (Agile) methodologies." - participant 15

> " I've never really thought of it that much of a conflict. I mean one of the things about data scientist is actually that it has to be adaptive." - participant 6

Another problem expressed by the interviewees is that in many cases the data set for a POC and a full-scale project will be different, For example, maybe it is not in real time or it runs on an older machine. Hopefully, a data scientist can reuse previous logic to some extent but there is a remaining question of how to develop and test a model for future uncertainties that up-scaling may introduce? There is a risk involved according to the agile sprint cycle and the majority of the interviewees, taking it to step by step is one way to mitigate that risk. The other interviewees prefer to have a longer time plan set from the beginning.

4.2.2.2 Solution for conflicts

When starting a ML project, one solution could be to try to split the machine learning process into different areas so someone specializes in working with refining and another in building the model. There are no well-established principles in data science like in software engineering so most projects are unique in terms of the working method. Also, working with quick deployments requires infrastructure that supports short iterations which can be tricky since the support structure of software engineering is different compared to data science since the data is a moving variable. This leads to the difference in software maintenance and ML maintenance where the later requires more specialized competence.

Another reoccurring tip was to bring in a data scientist from the beginning in order to scope the project: Is the project feasible? Furthermore, how experienced should the data scientist need to be in order to determine the potential of the data is hard to determine at the beginning of the project.

4.3 Software engineer

In this section, the role of a software engineer is explored together with traditional software engineering challenges that also may arise in ML projects.

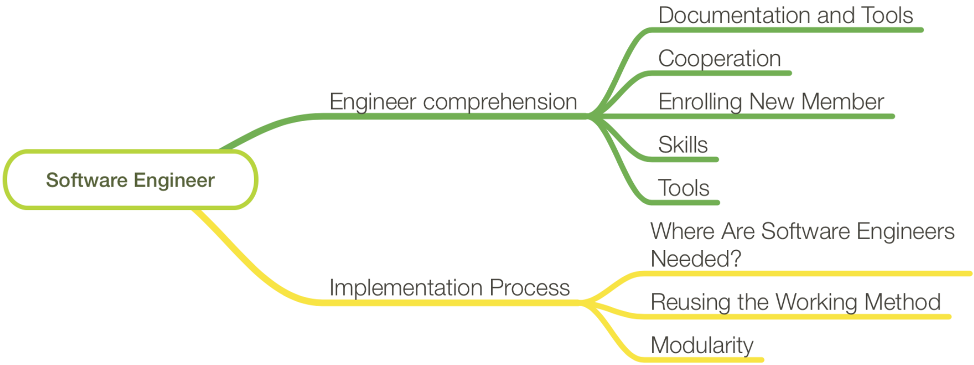


Figure 4.5: Result Mind Map for software engineers

4.3.1 Engineering comprehension

All interviewees talked about the importance of better communication among team members to understand the ML problem and increase engineering effectiveness and by different tools and working methods. Challenges exist such as how to enroll a new member faster, effective and humane; how to collaborate, how to deal with the conflict of different roles in a team, for example, between data scientists and software engineers. These will be answered in this section.

4.3.1.1 Documentation and tools

The tools and techniques used in the machine learning area sometimes are not documented well compared to the software engineering area because of the rapid emergence. No documentation on how to run the tool, no exploratory data analysis files and no reference points make it really difficult for engineers to understand. This makes it hard for data scientist and software engineers to understand what the output should be and how to connect it. Based on the interviewee's opinion, there should be more documentation for machine learning tools, framework, and libraries. One good example of comprehensive documentation is a tool called Rapidminder. Some interviewees also suggest using cloud services for data storage and computational power since it can scale up well when needed. Apart from this, engineers are advised to learn how to use Pickles, Spark, Scikit-learn and other packages in order to save and run models.

For machine learning project itself, sometimes there is no off shelf documents for project details, most case the team only use presentations to understand. In a machine learning project, one interviewee mentioned that documenting what is being done for pre-processing can help the understandability between data scientists and software engineer, even for clients. Also document the clear separation of phases such as the purpose of the machine learning models, the input and extracted output, the purpose of the engineers and so on. One good example of documenting and managing tasks told by one interviewee:

> "We have documentation in the code, we actually also use tools for managing tasks so you can always look at what has been achieved right. We make sure that we keep version control working. I think that is the very best way." - participant 10

Furthermore, one of the goals of documentation is to reduce bias according to one interviewee, which means to keep none obvious biases for all method and lead to further non-obvious biases for the whole project. Transparency and self-explaining code lower the risk of projects since developers might change during a project's timeline.

4.3.1.2 Skills

All techniques have their fields of application. When it comes to machine learning, one question often asked and also described in subsection 4.1.2 is where do the problem-solving start. In university, it is common to have a technology and researchers are looking for a problem to fit it, while in industry, on the other hand, the engineers are seeking a solution from a problem. Therefore, developed skills that lets engineers dig deep in understanding of a problem before selecting the technologies is important. This predicament also exists for data scientists described in subsubsection 4.2.1.8.

4.3.1.3 Cooperation

One interviewee recommended that meetings and workshops organized by data scientists to explain their work and method can make everyone on the same schedule. Companies also do weekly stand-ups together including data scientists and software engineers, speaking about their work. One interviewee stressed that deep learning engineers or data scientists are core located for frequent discussions.

For the newly enrolled engineers, team members can help the person to find related work to look at. Besides, one interviewee mentioned that the project team has all access ready when they start, so the newly enrolled engineers don't need to set up anything themselves. Another aspect of cooperation is version control tools for the code, data, and documentation, tools for managing tasks is also used.

4.3.1.4 Enrolling new member

For newly enrolled members, most interviewees suggested first to go through some examples in the repository and go through the source code factory, for example, to understand the Dev-Ops pipeline.

> "For the softer part, I think the only ways is to jump into the cold water and just try to hack and well, learn by doing." - participant 9

As the interviewee said above, new members should run the code and see what happen. They need to learn the software and learn the theory. They need to have that required code abilities like using PyTorch. This could be verified by giving the person different tasks that lead to an end to end knowledge. For example, let them take one feature from the source to the dashboard.

At the same time, the manager or the old employers will use PowerPoint to explain the project, sometimes they may use the same presentation used for customer, to help them understand the background, value, common goal for the project. New employees should ask for a better understanding since they need the full perspective of the project.

In a big project, using different containers could break down the solution and engineers can explain the purpose of each part to the new members and how it contributes. The old employees describe the input and output of the project and hear the new coming engineer's first impression, they may have some new insights. But old employees should understand the new engineers need some time. One interviewee expressed that, typically the time for the new members to get fully assimilate into the group might be a week or longer, depending on the complexity of the project. They get to know the project gradually from meeting to meeting, from workshop to understand what is going on, what should they present to clients.

4.3.2 Implementation process

With data science becoming more widespread, more and more software engineers will be involved in ML projects according to the interviews. This section covers where software engineers are needed and how modularity plays a part in making the ML projects more accessible.

4.3.2.1 Where are Software Engineers needed?

All interviewees believe that software engineers can help in structuring and developing data pipelines. For this type of development, software engineers are equally suitable since it does not involve any data evaluation. However, some of the more straight forward initial data science activities involving data cleaning such as clustering can be done by software engineers by using already existing packages.

> "(before starting a ML project) There are already so many problems in the company on preparing the data pipelines and streamlining the data." - participant 2

Furthermore, software engineers could help in projects where the code needs to be ported to other languages such as from python to C# in order to make it faster or to make it compatible with a proper UI. One interviewee emphasizes that when changing the language in a ML project it is important that the developer understand the algorithm being used in order to implement it correctly in the new language.

In addition, some interviewees emphasize that the client sometimes need to provide additional access and support in order to get the project up and running. Support in terms of API access, insight into their manufacturing, service or technology. This might be sensitive information but allowing data scientists or software engineer to access key features is important in order to have the possibility to determine the quality of the data which ultimately decides the future of the project. This dialog with a customer can be conducted by a software engineer as well as a by data scientist depending on who has responsibility for the initial ML activities.

However, a drawback to introducing software engineers in ML project is the communication problems that may arise. There may be a challenge in terms of technical language since there may be misconceptions about how the model works etc, as described in subsubsection 4.3.1.2 and subsubsection 4.3.1.3.

In addition, three researchers expressed that these POCs can usually be done without much support from software engineers. Stating that it is not until the scaling stage that more experienced software engineers and business people need to be present.

4.3.2.2 Reusing the working method

Half of the interviewees talked about how to reuse a pilot study. For example, knowing what type of input data and how the modeling process the data could be used in the next project. Once this working flow is created with ingestion and the parsing to plug in the new model, there is no need to rebuild everything. Especially for general projects, it is important to define the input and output of features and make some documentation in order to make it accessible to other researchers and practitioners.

4.3.2.3 Modularity

One question is the strategy of building the program around the ML model or building the ML model to fit a certain system. If the program is built around the model, there might be a problem with scaling. For example, if in the future there is a need for another algorithm, additional work is needed in order to rebuild the program to fit the new model, which is time-consuming. To tackle this, some interviewees suggest that a team should try to build a framework that can support additional models from the beginning. Participant 3 also emphasizes to implement the ML component separately from any kind of software in the purpose of having low cohesion between the parts as described by the quote below.

> "I would try to implement these components or machine learning components separately from any kind of software that you are building." - participant 3

One interviewee specifically said it is preferable to use containers such as Kubernetes[3] in order to isolate different parts. In addition, the sliced architecture makes it easy for the new person to contribute and understand (see subsubsection 4.3.1.4 for detailed describe), even if it is just simple tasks. Another positive aspect of Kubernetes is that it also holds a lot of small microservices that can be useful when scaling a ML project.

4.4 Management

In this section, topics related to high-level decisions and their possible effects are presented.

[3] Kubernetes is an open-source container orchestration system for automating application deployment, scaling, and management.

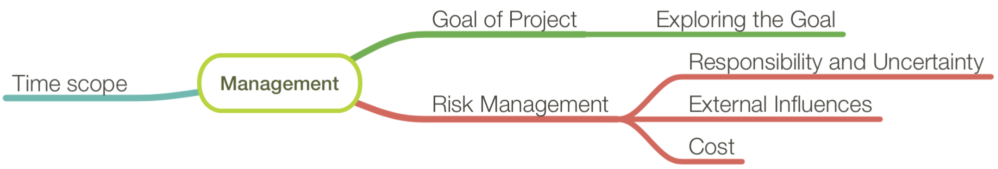


Figure 4.6: Result Mind Map for management

4.4.1 Time scope

All interviewees express that predicting the required time for a project is hard. In early POCs most researchers agree that a set deadline of three months is normal. However, with the disclaimer that it is the quality of the data is what really determines the required effort for a restrained three-month time scope. A POC contributes with a minimum quality check, letting the data scientists explore the data, therefore, the data scientists can determine if it is usable for the continuation of the project. The exploration phase also depends on the experience of the data scientists, which is a combination of the skills of the person as one interviewee stressed: the more a data scientist understand, the easier it is to make a good time estimation.

> "We're not the best at doing time estimations. I think we generally underestimate the effort it takes to try out new things... - participant 10

In terms of time scope, there can be a difference between academia and industry since in industry the customer may assign the level of the desired accuracy, and the freedom to choose their own path for data scientists and software engineers is limited. While as a researcher, the quality of the classification might be the most important thing since the only customer is the researcher and the university. For example, developing new algorithms with trial and error is very time-consuming which might not suit POCs.

The general opinion of the interviewees is that two to four weeks are needed to do an initial analysis of the data but that building a data pipeline, getting the data in the right format is time-consuming and should not be taken for granted. One interviewee who coming fresh out of university expressed that the customer data is usually not ordered in neatly CSV files as a minor shock, realizing that formatting the data and building the pipeline is a big part of the data scientist work.

In bigger projects such as governmental projects, the time plan is usually longer, for example, 5 years. One interviewee said that in these types of projects it is common to write a project plan and a summation of every year in order to keep the funding. This takes time but it also provides freedom ones completed, freedom to explore model options. One important result was also that the timing of the project can have a huge impact.

> "For governments project, is required to make a proposal, also to make projected plans every year and guess the sum of the year we are accepted by the government. The client project can be long. But each contract unit is not so long, maybe a half year. Usually when new contracts are, renew a short contract multiple times, is the way that we can avoid as a risk.

4.4.2 Goal of project

Specifying the goal of potential collaboration is the starting point for many data science projects according to the interviewees. Below some of the challenges that may arise in the process.

Eight people specifically mentioned the importance of making the goal of the project clear. Before looking at the data, it is important to point out the goal of the project. For example, according to the interviews, some companies have started to gather large data lakes (since they have come to the conclusion that data) while not having an understanding of what to do with it, hoping for miracles without a clear view of their own resources. Therefore the initial focus from a data scientist should be to evaluate the client's expectations. After the initial evaluation, the data scientists could assess the client's data in relation to their expectation. Jumping straight into the data lake is unadvised since it is hard to make sense of it without a clear goal. For example, data gathered from expensive sensors may not be equivalent to good measurements of quality in a product. It is important to know what the data should be representing and why before investing in expensive data gathering equipment according to the interviewees.

> "You get surprised in how much time you need to put into disclaimers in the beginning. If I could I would put even more time to handle the expectations. What are the requirements of the company?" - participant 15

4.4.3 Risk management

Risk management is an important task for leadership in companies. According to all the interviewees, this is highly relevant in the ML field since a large part of data science is about providing insight that can help people make better decisions. If customers are not aware of the potential shortcomings of a ML recommendation, they might end up making blind and harmful decisions.

> "The analytic objective needs to evaluated, is ML even the right way to approach the problem?" - participant 15

4.4.3.1 Responsibility and uncertainty

Eleven data scientists from the study agree that data scientist has a responsibility to convey the risk of the model. In for example anomaly detection, there is usually a probability distribution incorporated in the model output. This is by the fact a distribution that a span of possible outcomes with more and less uncertainty. Worth noting is also the fact that two of the interviewees expressed that they thought it was ultimately the user's responsibility. Stating that data scientists cannot make the final decision for the customer but to provide the best possible recommendation, letting the customer make an informed decision. Another interviewee stated that data scientists should sign a contract renouncing the future responsibility of the model prediction that is mostly a public relation issue.

> "I think a customer who blindly trusts things should not be one. I mean risk management is the prime task for leadership in companies right." - participant 1

Sometimes data scientists do not have time to explore all possible scenarios of a data set. Hence, a local peak might be found in the result but not the optimal global peak. In this case, data scientists could provide this information to the customer, letting them make decisions accordingly. Additional lab tests are needed to confirm exactly what the right approach is for things that might have a big impact. This is related to not overselling the solution but show the potential pros and cons of the model. This is sometimes contradictory since humans tend to dislike approximations and management want clear cut recommendations. However, the full picture should be presented to the customers if they are going to be able to make an optimal decision. A data scientist should combine human inspections from external sources such as domain experts together with the model output in order to increase the security. For example, in the stock trading market, one safety measure could be to not let the model buy stocks itself, only providing recommendations for humans. At the same time, keeping in mind that humans are not perfect either, people have biases. Another method in order to increase the predictive validity is to combine multiple models, predictions and test are described in subsubsection 4.1.3.5.

4.4.3.2 Cost

In order to further increase the explainability of the solution, the data scientist should convey the potential effect of the model, for example, what does this recommendation mean in terms of financial saving or cost for the company, etc. Other effects include what is the cost of risk, cost of one miss-prediction, the alternative cost for not choosing the ML solutions. For example, one interviewee mentions that in the area of cancer detection, false positive is much less risky than a false negative. This is crucial in adjusting the sensitivity parameter of an algorithm since it could potentially save peoples lives. What is the cost of a false positive and false negative prediction? Another interviewee expressed that this type of packaging of a model can be handled by another business unit where they compute the financial risk in regards to requirements, expectations, and features of the problem. In that way, the data scientists can focus solemnly on increasing the quality of the model. Another reoccurring theme related to risk management is to start the project with simpler common models in order to quickly deliver value to the customer, reducing the risk of complexity. In addition, another important aspect of mitigating risk is as stated in earlier chapters (subsubsection 4.2.1.1) to conduct conservative pilot projects.

> "If I have a smaller number of models that are good enough I may actually get into the range where my solution is economically viable. As opposed to, you know technically optimal and that balance between what is economically viable, the cost and then the quality of the solution. This is what we call engineering." -participant 6

4.4.3.3 External influences

ML field is continuously under development. Some interviewees thought that reading about these external influences, keeping oneself updated from articles, blogs, etc were crucial in their day to day work. Data scientists rarely develop anything from scratch, instead, it is common to rely on existing systems and libraries, which means that there is no point in hugging a specific solution if there is a new better method accessible. At the same time, several interviewees point out that it is wrong to trash previous efforts, instead, try to integrate it or reuse it in another project.

> "We don't trash our own efforts because I think it's always a mistake to do that. We try to integrate that new idea into what we have been doing. We can always modify our pipeline. We can add or remove components as we see fit." - participant 10

Most interviewees express that trying out a new technique usually does not require extensive amounts of time but can be done rather quickly. Of course, more complex models require more time and resources to evaluate. But according to the interviewees, the most time-consuming part is actually to figure out how the input should be fitted into the model, in other words, how the data should be parsed and how to transform and formulate the clients' problem in a quantitative way. Another reoccurring thought is to be sceptical of hyped techniques since it might be hard to adapt them to another specific problem.

> "If the methodology is like big success then I would be first sceptical of it because I know that it not like magic..." often is not the case that a method works well for other tasks can be applied directly to my task or to my project. So it needs some adaptation and some work. So usually I would be sceptical or conservative about the fixed tool to have sharp change or transition." - participant 5

4.5 Client

In many ML projects, there are different stakeholders involved. In this section, explainability and economical aspect are presented from a client's perspective.

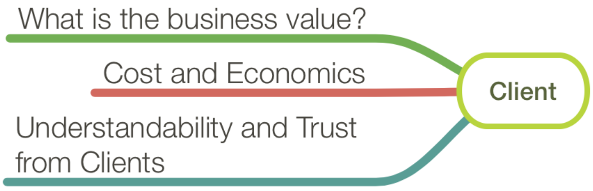

What is the business value?
Cost and Economics
Understandability and Trust from Clients
Client

Figure 4.7: Result Mind Map for client

4.5.1 Understandbility and trust from clients

According to nine of the interviewees, translating the technical aspects into business value and building trust with the client are the most challenging parts of ML projects.

> "It's the people part that's the hard part." - participant 15

The client wants to know what the business value of the output of an algorithm is. Therefore, data scientists need to make people trust their model or their idea. For example, according to one interviewee, data scientists can not show machine learning validation game charts to a business person but is required to translate the output in the client's own natural language while being honest about its faults. There are different ways to validate a model, for example, a technical evaluation which solemnly focus on the prediction rate, or a series of measures of business impact. Different types of evaluation criteria need different explanations in order to make the client comfortable in the models. For example, 95 percent prediction accuracy can be great in one small project but unusable in a larger project.

> "It's tempting to oversell but you need to make sure there are no legal consequences if mistakes arise." - participant 9

Another common experience is that clients want the most complex models, e.g. a deep learning model. 9 Interviewees stated that this is usually rooted in a lack of machine learning experience. There is a possibility that simple linear regression analysis is the easiest, quickest and best way to go. Furthermore, perfect performance and detailed explanation of the algorithm is usually not required, it is more about showing that the model can predict something valuable to the client, visualizing the usability and output in simple terms, which naturally is easier with simpler models.

> "In smaller projects, it's always about visualizing everything very nicely and structurally. So it's not about the model but more of like the usability and understandability." - participant 11

Ultimately, all interviewees agree that reaching a business consensus for the continuation of projects largely depends on the data and how well the model prediction output is translated to usable insights for the client. However, some Web-based companies help in this process by doing A/B testing in order to evaluate the business value. It is naturally easier with younger companies who were raised in the cloud of ML era.

4.5.2 Cost and Economics

All interviewees mention the challenge of balancing resources and model output quality, specifically the data scientist consultants from IBM emphasized the importance of having a paying customer. The solution needs to be economically viable as opposed to technically optimal. The data scientist needs to take in consideration external parameters related to business value: what it the cost of a repair? How to balance between cost, accuracy, economics? All these are unique for most application, and higher target prediction accuracy usually comes with higher costs. The interviewees also mention that in industry, clients only pay for coding, not for the learning processor for a better comprehension of the problem and choice of algorithm, which is essential for data scientists.

4.6 Practical verification result

As mentioned in section 3.4 and Figure 3.8, the implementation process including preparation in terms of data and the problem, implementing the models, and the final results will be presented in this section.

4.6.1 Understand data and preparing the data pipeline

Plant K whose sensor structure is shown in Figure 4.8 was the chosen project of Energy Machines due to its size, accessible history data, and relevance in terms of known failures. (The plant name is left undisclosed due to privacy aspects.) Several meetings were arranged with engineer D and architecture L in Energy Machines in order to set the basic understanding of the system, structure, and data. The system had a pre-written API in Swagger[4] for extracting historical data, alarms, and information for projects and sensors. However, it was not possible to use the API directly since the required data format needed to apply the ML models was different from the data format given by the API. After explaining this, Engineer D assisted in providing the correct data format by updating the API to provide the necessary data extraction options. The post request running in Postman[5] could then return a CSV data file in the right format after authentication from the new API.

The next step was to clean the data and since the researchers did not have the authority to rewrite the API of Energy Machines which connected to their database, and the new API offered by the engineer in Energy Machines was not adaptable for every case. The researchers manually worked on the data, formatting such as changing the data time format from million-second data into "yyyy-mm-dd-hh-mm", removing unused data columns as well as identified and replaced bad quality data which would benefit the further clustering work according to the interviews.

4.6.2 Clustering

Clustering is commonly used in unsupervised learning in order to help structure the data. Furthermore, clustering data makes it easier to look for further similarities, e.g, in anomaly detection it can be used to identify points outside of the main data distributions, hence an anomaly. Another aspect is that the prediction could get

[4] Swagger is an open-source software framework that helps developers design, build, document, and consume RESTful Web services.

[5] Postman is a complete API development environment for API developers.

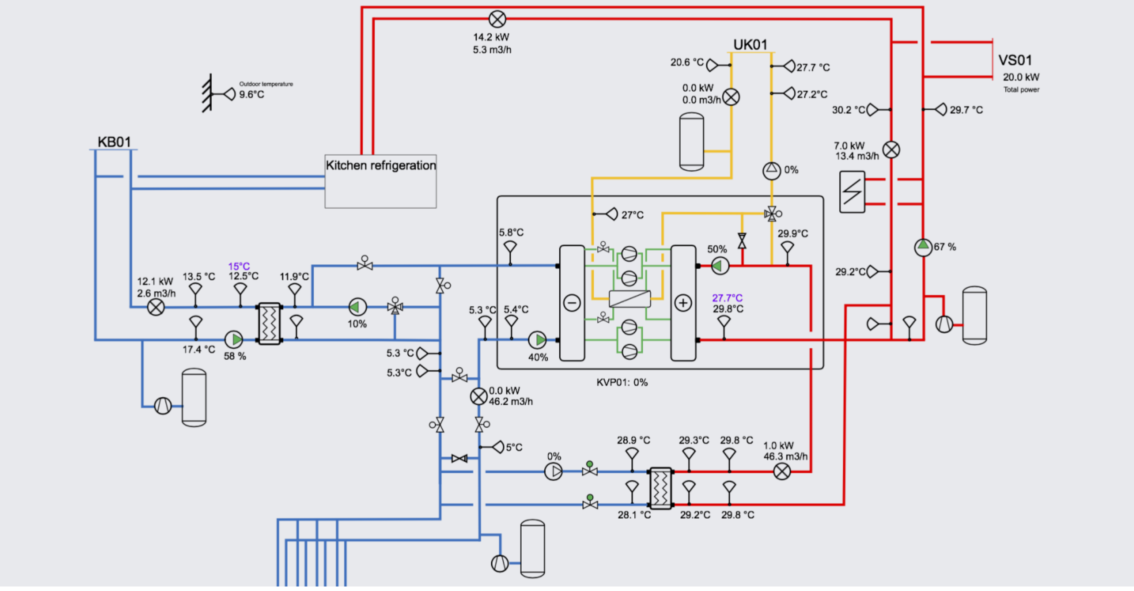
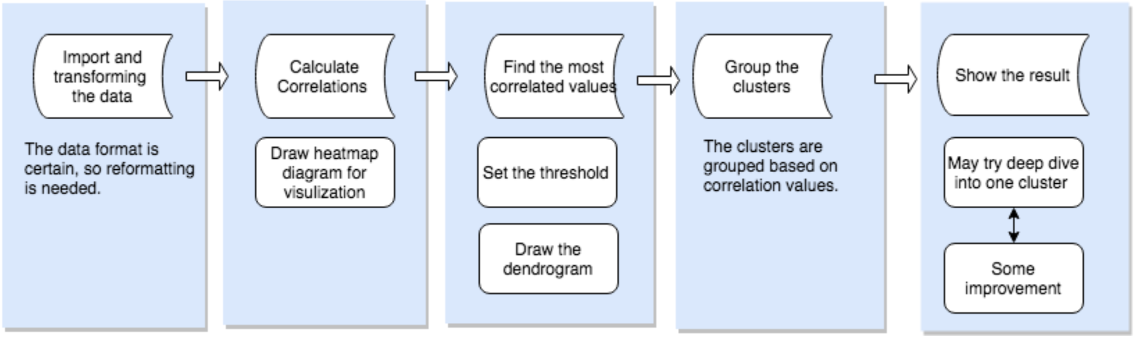


Figure 4.8: Overview of system and sensors in Plant K

more accurate results inside clusters since in this case, the sensors inside the same cluster have similar behaviour and may influence each other. Energy plant K has some physical correlated sensors (based on domain knowledge) which according to the interviewees should be prioritized, such as sensors placed on a single module. As we mentioned in subsection 4.6.1, meetings were held with engineer D in Energy Machines in order to determine the optimal sensors for a physical cluster. Below in Figure 4.9 a visual structure of the data cleaning and structuring process is presented.


Import and transforming the data

The data format is certain, so reformatting is needed.

Calculate Correlations

Draw heatmap diagram for visulization

Find the most correlated values

Set the threshold

Draw the dendrogram

Group the clusters

The clusters are grouped based on correlation values.

Show the result

May try deep dive into one cluster

Some improvement

Figure 4.9: Steps conducted in the clustering process of the practical verification.

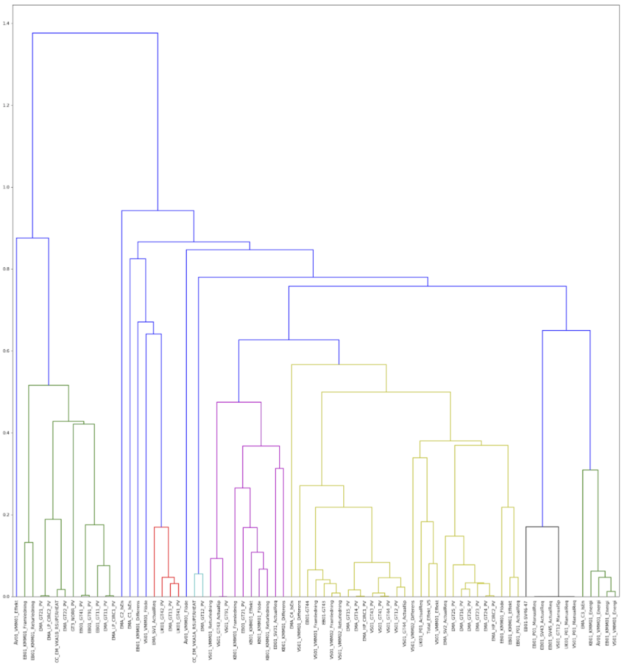


Figure 4.11: Dendrogram of the identified Correlations

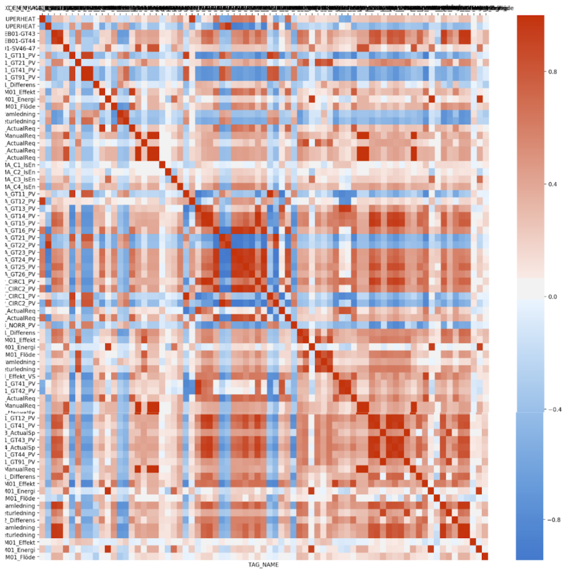


Figure 4.10: Heatmap of the identified Correlations

Clusters are made based on correlations of sensors. After the cleaning and formatting of the data, correlations between each pair of sensors were calculated. A heatmap was drawn as shown in Figure 4.10, increasing the understandability through visualization which several interviewees mentioned as described in chapter 4. Then the sensors were ranked based on correlation values for further calculation. To conduct the clusters, a dendrogram(shown in Figure 4.11) analyzing the taxonomic relationship is calculated and the threshold index was adjusted several times to make more reliable clustering result. The number of cluster group is 13 in the first iteration. The result would be changed during the iteration period described in subsection 4.6.4.

4.6.3 Hyper parameter tuning

Hyperparameter tuning is also called hyperparameter optimization ((J. Bergstra, Yamins, & Cox, 2013)). Tuning of hyperparameters is an important part of understanding algorithm performance and should be a formal and quantified part of the model evaluation. There are a number of available algorithms used for tuning: sequential model-based global optimization, Gaussian process approach, and tree-structured parzen estimator approach(J. S. Bergstra, Bardenet, Bengio, & Kégl, 2011). Considering our algorithms, Isolation Forest algorithm contains one parameter, and Dynamic Boltzmann Machines algorithm contains 5 hyperparameters. In addition, the researchers are junior data scientist with limited experience of the different hyper tuning methods and as a result, decided to do hyperparameter tuning manually. The analysis and validation of the anomaly detection indexes were done by comparing the results to the leakage information provided by Energy Machines. The core hyperparameter is contamination level in both algorithms, it is used when specifying the proportion of expected anomalies in the training data set. The result of the final hyperparameter value will be presented in chapter 4.

4.6.4 Iterations

During the process of exploring the models of the two algorithms, the researchers had three iterations based on Energy Machines' and one senior data scientist's feedback. In the first iteration, 60 sensors divided into 13 clusters where identified. These were then presented to Energy Machines who emphasized detecting the leakage problem is their vital issue, rather than analyzing the whole system in plant K. They also presented the leakage problem again and added additional sensors to the physical sensor group. In the second iteration focus was put on the physical cluster only which including 11 sensors as shown in subsection 4.6.1 in Figure 3.11. To get a more precise result, we dug into the data trend by visualizing the data from recent years. It presented that the data depended heavily on temperature and seasons and after consulting with the senior data scientist form IBM, the input data was limited to only the winter period from October to April. Lastly, the data science expert from IBM also advised doing a second clustering inside the physical cluster, aiming to find more correlations and multivariate prediction result. This was the final iteration which provided the best prediction.

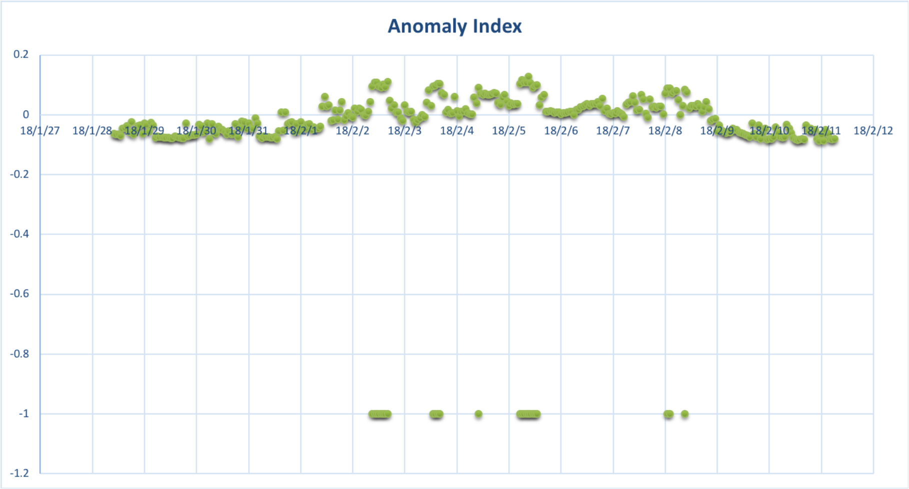


Figure 4.12: Detected anomalies in the winter of 2018

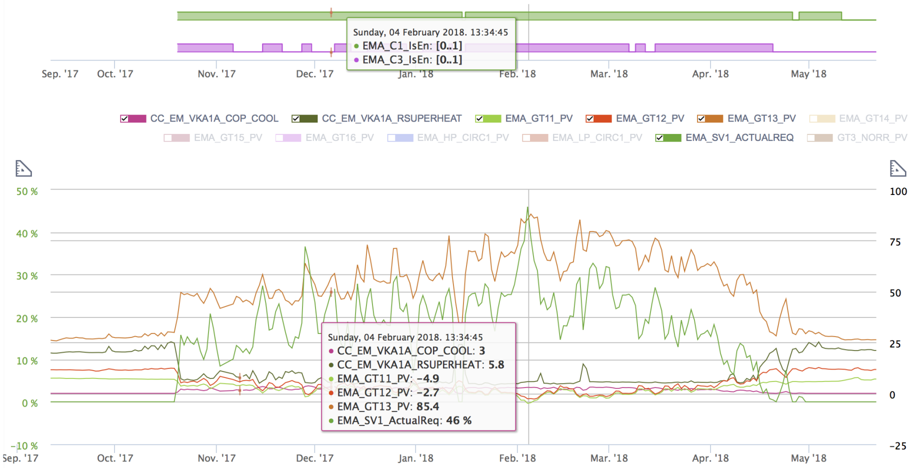


Figure 4.13: Detected anomalies reflected in real data for winter data

4.6.5 Final result, configuration, and feedback from client

The input data for training was organized by per hour and because of the seasonality, the selected time range was three winter: 2017/01/01-2017/04/15, 2017/10/01-2018/04/15, 2018/10/01-2019/04/15. The prediction finally spotted a group of anomaly points throughout three winter periods. As shown in Figure 4.12, the most detected anomaly period where 25 anomaly points and ranged from 2018-02-02 to 2018-02-08. These were suspected to correlate to the leakage problem in the refrigerant system which was later confirmed in the meeting with Energy Machines in the middle of April 2019. In addition, the real data chart is shown in Figure 4.13 also verified that there was an abnormally high value at 2018-02-04.

At first, the data cleaning and the hyperparameter tuning were started in March 2019 on all available data. Then, Energy Machines provided two time periods with suspected leakage problems. According to their service technician: one leakage was first discovered and fixed in 2016-12-01 (it had probably occurred on 2016-11-17, the service technician did not know for certain); the second leakage had been discovered and fixed at 2018-01-08 (it had probably occurred on 2017-11-15, the service technician was also uncertain about this).

Regarding Energy Machines history data, the starting date is 2017-01-17, as a result, the first leakage problem in 2016 was discarded since there was no available data in the system. Hence, we decided to narrow the data closer to the winter of 2018, focusing on the second leakage problem. We explored the data quality according to the interviewees' suggestions, e.g looked for missing values and max/min values and irritably changed the hyperparameters until a matching number of anomalies as suggested by Energy Machines occurred. We also got support from an experienced data scientist from IBM as to how to handle the missing values. However, after multiple weeks, the time period of our predicted anomalies did not match the suggested time period of the second leakage problem. We concluded that we needed more domain knowledge in order to make a better prediction. Hence, we scheduled a meeting with Energy Machines with the hope of extracting more details about their data and their system. At the meeting, we decided to draw the timeline of their system and visualized our outputted anomalies(shown in Figure 4.14). Since these anomalies did not align with the service technicians presumed dates, we tried to think of new ways to better the prediction. This resulted in a discussion about how the systems energy consumption hypothetically would be affected by a leakage, e.g if the energy consumption would go up and how that could be measured. As it turns out, thermodynamics is a complicated area but if we could capture the behavior of the systems SEI feature, we could possibly measure the energy consumption more thoroughly and hopefully see a clearer pattern.

After further discussion, developer D recalled the service technician responsible for plant K to ensure the reported dates where correct. As it turns, they were not, the corrected dates, also shown in Figure 4.14 stated that the leakage was detected 2018-05-14, which indicated the leakage must have happened before the middle of May

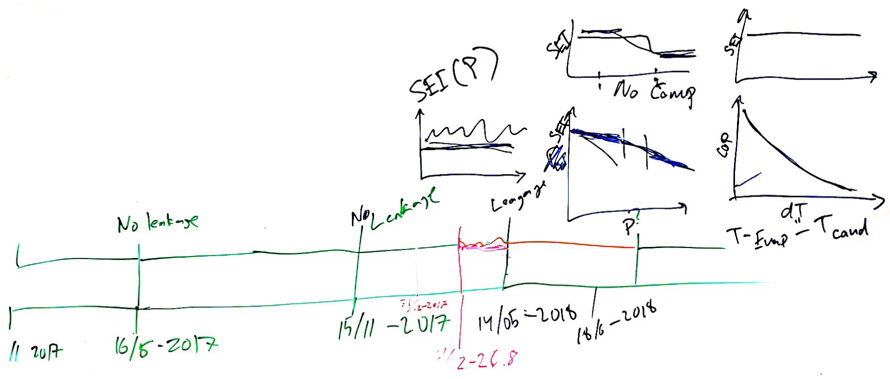
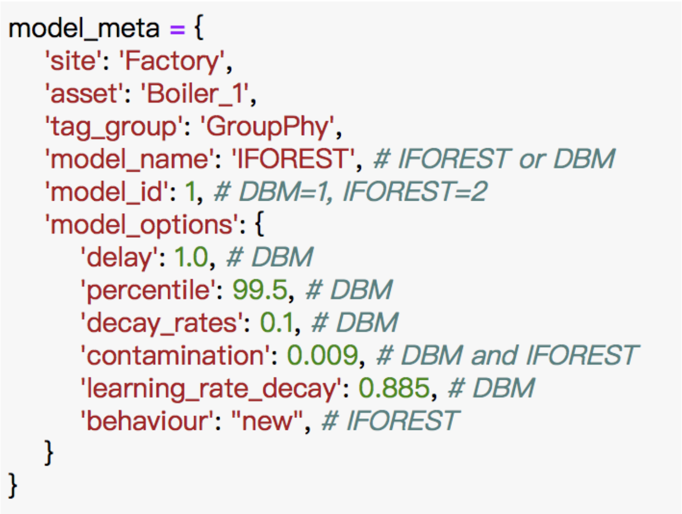


Figure 4.14: Meeting record of anomaly validation process at Energy Machines



```
model_meta = {
    'site': 'Factory',
    'asset': 'Boiler_1',
    'tag_group': 'GroupPhy',
    'model_name': 'IFOREST', # IFOREST or DBM
    'model_id': 1, # DBM=1, IFOREST=2
    'model_options': {
        'delay': 1.0, # DBM
        'percentile': 99.5, # DBM
        'decay_rates': 0.1, # DBM
        'contamination': 0.009, # DBM and IFOREST
        'learning_rate_decay': 0.885, # DBM
        'behaviour': "new", # IFOREST
    }
}
```

Figure 4.15: The tuned parameters used in the final models.

2018. He also said no leakage had been discovered during the autumn service check in October, the previous year. This updated information verified our prediction which meant that our anomalies, centered around 2018-02-04 were in the stated time frame. The next step involved a presentation of the anomalies to the domain expert at Energy Machines. He concluded that the discovered anomalies where indeed abnormal behaviour that should not occur in the system and was intrigued to further investigate the specified dates.

Final settings

Regarding the final dates for the anomaly prediction and the final hp parameters, is shown in Figure 4.15. The final cluster configuration is shown in **??** and the inner cluster is shown in Figure 4.17. In addition to the prediction on the winter data, we also trained the models by using all available data (three years of data). The results is presented in Figure 4.18 and points to an anomaly period during 2018-06-04 to 2018-06-19, which is reflected in Figure 4.19. These were also connected to the leakage since the repairing operator reset the system during this period.

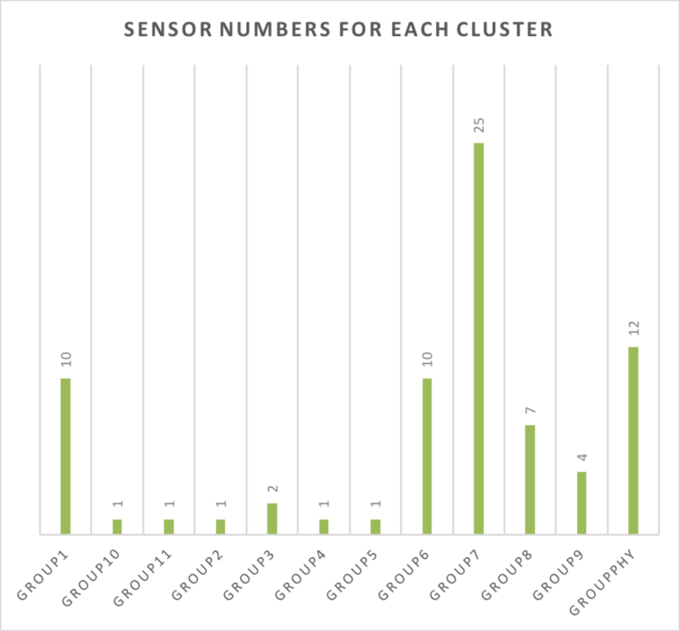


Figure 4.16: The final cluster group

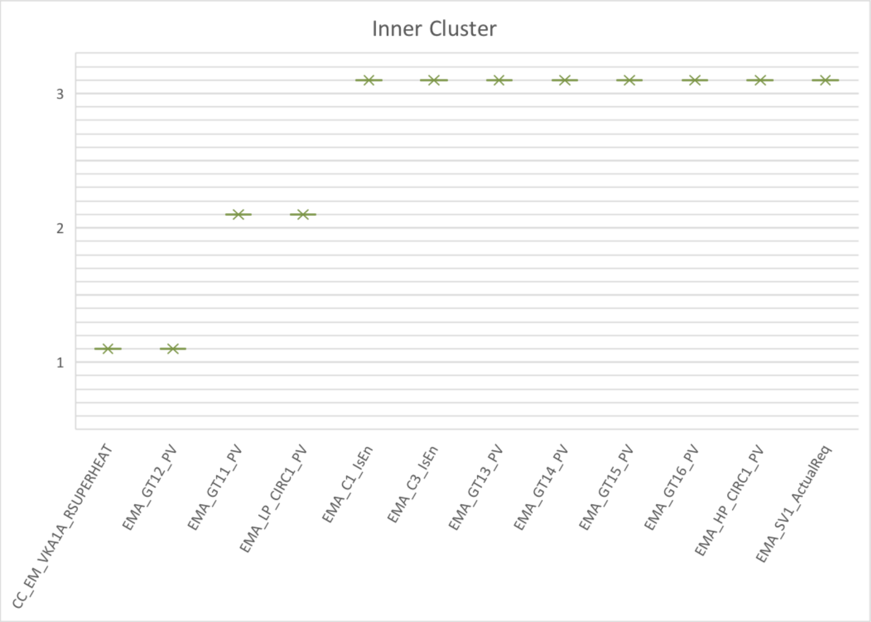


Figure 4.17: Inner clusters of physical cluster provided by Energy Machines domain expert.

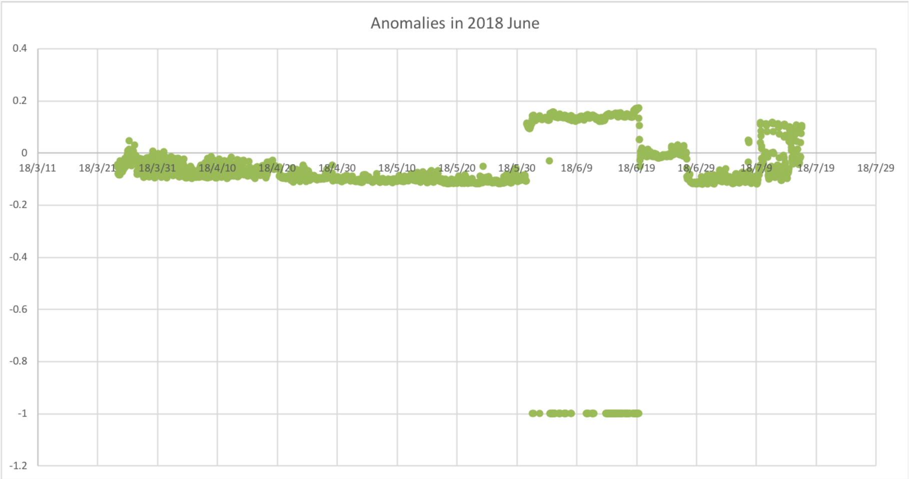


Figure 4.18: Detected anomaly result on all available data.

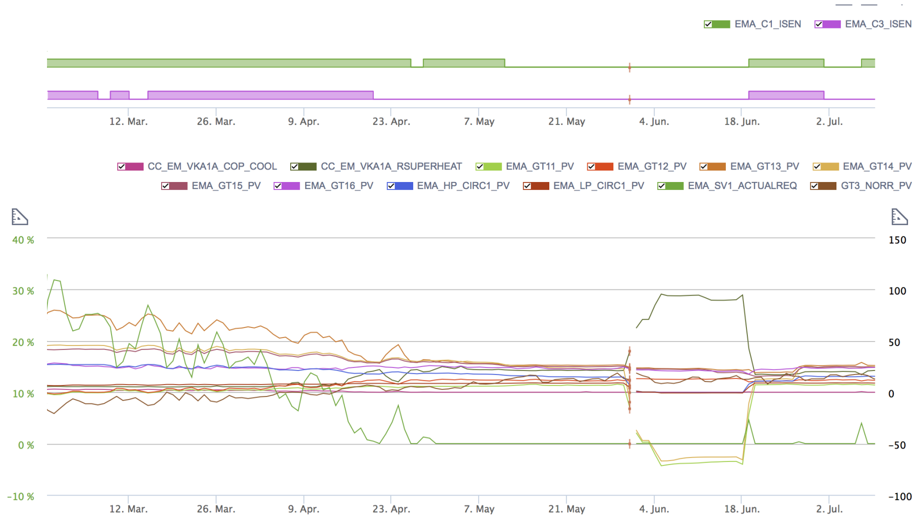


Figure 4.19: Detected anomalies reflected in real data in whole season

5
Discussion

In this chapter, the results from chapter 4 are discussed and analyzed. The discussion covers the main findings from the result chapter based on a number of mentions, their interconnections, and interviewee emphasis. It is structured according to challenges and strategies, lessons learned, future trends, a reflection of theory and threats to validity and each section covers the analysis of interview results with the corresponding experience of the practical verification test. An overview structure is shown in Figure 5.1 and Table 5.1 display the themes together with summarized descriptions and correlations to roles.

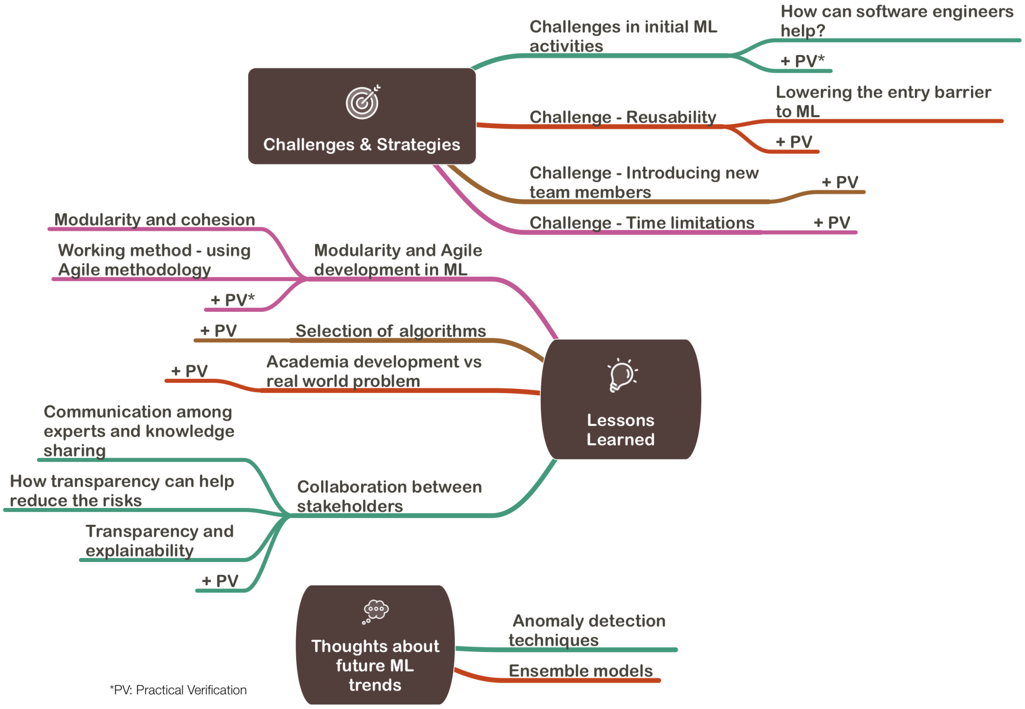

Challenges & Strategies
Challenges in initial ML activities
How can software engineers help?
+ PV*
Challenge - Reusability
Lowering the entry barrier to ML
+ PV
Challenge - Introducing new team members
+ PV
Challenge - Time limitations
+ PV
Lessons Learned
Modularity and Agile development in ML
Modularity and cohesion
Working method - using Agile methodology
+ PV*
Selection of algorithms
+ PV
Academia development vs real world problem
+ PV
Collaboration between stakeholders
Communication among experts and knowledge sharing
How transparency can help reduce the risks
Transparency and explainability
+ PV
Thoughts about future ML trends
Anomaly detection techniques
Ensemble models
*PV: Practical Verification

Figure 5.1: Mind map overview of discussion involving three main areas, corresponding themes and practical verification.

Challenges	**Description**	**Relevant Roles**[1]
Initial ML activities	The initial ML activities include setting up a data pipeline, cleaning data and other supporting activities	SE
Reusability	The challenge revolves around capturing ML experience in reusable formats with the help of e.g. transfer models and AI platforms	SE, DS
Time scope	Time is a limiting factor in almost all projects. In ML projects it may have an effect on preference towards simpler models.	SE, DS, CL, MG
Introducing new team members	There is a lack of structured methods regarding how to get members up to date in ML projects.	SE, DS, MG
Lessons Learned	**Description**	**Relevant Roles**
Modularity	Keeping parts of ML systems separate may increase the success rate of projects since ML is iterative and naturally dynamic.	SE, DS
Agile methodology	Traditional agile workflows are important in ML, especially aspects such as close customer contact but it usually comes with a price of model accuracy.	SE, DS
Selection of techniques	Simpler models may be preferable in ML projects where time, data quantity and computational power are restraining factors. However, there are cases where more complex models have shown great results such as in NLP	DS

[1] SE: software engineers. DS: data scientists. MG: management. CL: clients.

Matching academia and industry	There is often a miss-match between the two entities in terms of problem-solving. GANS is a good example of a technology that it is fascinating but hard to find an actual use-case for.	SE, DS, CL, MG
Collaboration with stakeholder	Domain knowledge is a central part of ML. In many cases, the project is dependent on collaboration between different stakeholders and their willingness to share information.	SE, DS, CL, MG
Future ML trends	**Description**	**Relevant Roles**
Ensemble models	Combining models outputs in order to address problems from multiple angels is one interesting future possible trend.	DS
Anomaly detection techniques	Anomaly detection is a relatively unexplored area in the industry, multiple approaches and algorithms can be used it will be interesting to see if there arises any particular standard method in the field.	DS

Table 5.1: Summarized challenges, lessons learned and future ML trends

5.1 Challenges and strategies

In this section, the main identified challenges with ML implementation is presented together with possible solutions and practical implication.

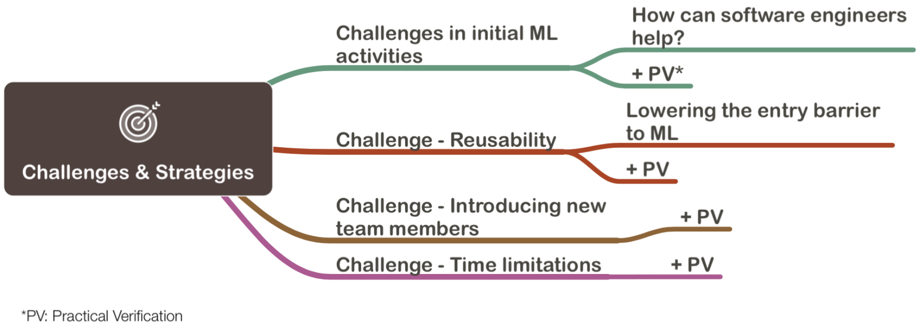

Challenges & Strategies
Challenges in initial ML activities
How can software engineers help?
+ PV*
Challenge - Reusability
Lowering the entry barrier to ML
+ PV
Challenge - Introducing new team members
+ PV
Challenge - Time limitations
+ PV
*PV: Practical Verification

Figure 5.2: Mind map of challenges and strategies

5.1.1 Challenges in initial ML activities

As presented in subsection 2.3.1, Saleema Amershi (2019) discussed the different stages of ML projects and three parts of the AI domain that distinguish it from other software engineering domains. An important aspect being that managing and structuring the data needed for ML applications is a complex and difficult process compared to regular software engineering activities. This resonates well with the results in subsubsection 4.3.2.1 and subsubsection 4.2.1.3 that present the initial activities in ML projects.

5.1.1.1 How can software engineers help?

The initial activities of implementing machine learning algorithms require a certain amount of data science knowledge, for example, choosing which method to use to replace missing values in a data set. This can be a more difficult and challenging task than regular software engineers perceive. However, as the subsubsection 4.3.2.1 show, we believe that software engineers can provide support in the earlier phases even with limited data science knowledge, primarily in the data gathering phase which usually requires setting up the data pipeline. Many existing algorithms and systems depend heavily on a strict data format that differs from the clients' data. Extracting data, formatting data through for example adapting APIs, is not the primary tasks of a data scientist who should instead focus on more specialized tasks such as cleaning, feature extraction, and model evaluation. Setting up the data pipeline and parsing data are initial steps that software engineers can help with.

5.1.1.2 Practical verification on initial ML activities

That the initial ML activities may be challenging was also confirmed through the practical implementation carried out (see section 3.4). A large portion of the practical project's time with Energy Machines went to waiting for the right data access and updating the client's data API to support the required data format. Two activities were reliant on the client's help. Another time-consuming activity was identifying missing values and figuring out what method to use in order to replace them. These steps required domain knowledge of Energy Machines system, i.e what happen to the missing values, as well as ML knowledge about which algorithms and methods to be used for replacing missing data. Replacing missing values may seem trivial to an experienced data scientist while from a software engineer's perspective it can be challenging and time-consuming. However, this does not mean that it is not possible for the later to complete the activity. Therefore, it is not unrealistic that in the future, more software engineers will work with and contribute to these early phase ML activities.

5.1.2 Challenge of reusability

One main challenge is the uniqueness of customer problems in the industry. Largely because ML models are heavily data dependent and unique data leads to unique problems. One possible solution to tackle this, according to the interviewees, is transferable models as stated in subsubsection 4.1.1.2. However, there is a discrepancy between theory and practice among the interviewees, since many believed that transferable model is a good solution to the reusability problem in theory, but few seem to have done it in practice successfully. Hence, in theory, transferable models could be a fast way to start up a new project but so far it seems to be too complex for the average data scientists. Furthermore, the most promising successful attempts have been done in vision and language recognition area by reusing the lower layers of a convolution neural network (see subsubsection 4.1.1.2).

5.1.2.1 Lowering the entry barrier to ML

In subsubsection 4.1.1.1 some interviewees mention the reason that the number of experts who can distinguish usable data and dig deep into its implications is relatively scarce. The reason being that the computational power needed to make use of data has been unavailable until recent years. However, according to interviewees, universities are slowly catching up to the growing demand. Still, some IT-giants are working on their own solutions, such as AI-platforms as exemplified in Figure 2.3. These platforms are made in order to capture data science expertise in an asset that lets people use ML at a lower entry cost and to provide easier access to scalable ML solutions. Some data science tasks can be tedious and time-consuming as expressed by the interviewees in subsubsection 4.2.1.3, this could indicate that there is a demand for automated services that simplify the daily work of a data scientist. For example, in supervised ML you need to build a reliable ground truth in order to have validated models. This can be expensive to do from scratch. If an AI-platform can provide support in this challenge then it could potentially provide significant value. Another positive aspect could be that the platforms lower the entry barrier to ML so that, for example, regular software engineers can use them and by that, bridges the gap between regular software development and ML in industry.

5.1.2.2 Practical verification

For the practical implementation test, IBM provided an anomaly detection tool. Not exactly a complete platform as described in subsection 2.4.1 but the tool did contain two moderately easy to use algorithms with supporting structures. This helped us in clarifying the specific input data required from the client and accelerated the start-up phase of finding and adopting a suitable algorithm which would otherwise have taken valuable time from the project. Hence, our practical case study confirms that at least some parts of AI-platforms/tool can support even junior software engineers in their work.

5.1.3 Challenge of introducing new team members

Enrolling a new member to a ML project comes with many of the same challenges as in regular software development according to subsection 4.3.1. The typical answer consisted of keeping the system documented, version controlled and walking through the system bit by bit. The main thing interviewees agreed on was that there did not exist a well-established method for on-boarding new members. What stood out from the trivial answers was that a few interviewees mentioned assigning a small and easy task in an end-to-end format. That way the person got eased into the system with a sense of contribution. In ML that could mean taking a data point from the data collection phase to the model training phase as visualized by Saleema Amershi (2019) in Figure 2.2. It would be interesting to research the subject more thoroughly in the future and investigate if there are any well-established methods.

5.1.3.1 Practical verification

Getting up to date with IBM's anomaly detection tool can be related to enrolling a new member in a ML project. Seeing a high-level overview and looking at requirements was the main supervised introduction, looking at the code and trying to run the different parts was however more useful. The main resonating idea from the interviews was to try the end-to-end format with a couple of data points. Our experience is that this, in many cases, is easier said than done and for us, it was challenging. However, once the required data format was solved other things fell into place but the feeling of accomplishment and contribution as mentioned by interviewees was nonetheless true.

5.1.4 Challenge of time limitations

Estimating the time-scope for ML projects is challenging according to the interviewees in subsection 4.4.1. Time restrictions introduce stress to the project which might encourage the data scientist to cut corners which ultimately affect the quality of the model performance. This might be acceptable if the performance meets the requirements of the customers, however, it is still worth questioning if this is a sound development for the data science field as a whole.

5.1.4.1 Practical verification

Time is a restraining factor during a thesis project. Structuring, limiting and evaluating the scope of the project was challenging, especially since different stakeholder saw value in different aspects such as academic height vs model output quality. Having an iterative dialogue with the stakeholder throughout the project helped in conveying reasonable expectations and ultimately served as risk mitigation. The time restrictions did affect the quality of the two models negatively but since the time-framework was set from the beginning it was well thought through strategy. Working as a data scientist consultant with customer expectations will probably always limit the scope of the project but as long as the time is enough to produce value for the client, it will probably not change.

5.2 Lessons learned

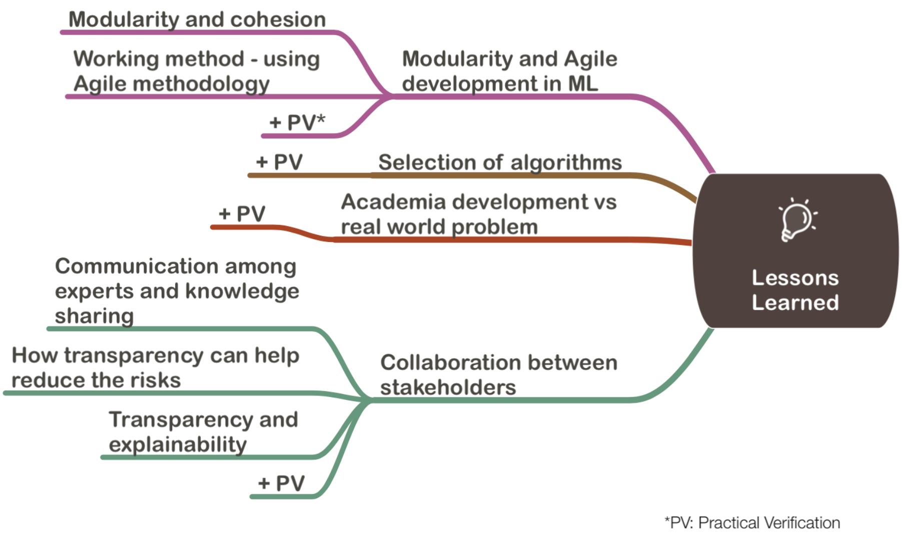

Lessons Learned
Modularity and Agile development in ML
Modularity and cohesion
Working method - using Agile methodology
+ PV*
Selection of algorithms
+ PV
Academia development vs real world problem
+ PV
Collaboration between stakeholders
Communication among experts and knowledge sharing
How transparency can help reduce the risks
Transparency and explainability
+ PV
*PV: Practical Verification

Figure 5.3: Mind map of lessons learned

5.2.1 Modularity and Agile development in ML

In this section, factors that could increase the success rate are analyzed regarding the perspectives from the project itself, the working method and techniques, people.

5.2.1.1 Modularity and high cohesion

In subsubsection 4.3.2.3, it is emphasized that increasing modularity is preferred in ML since the flexibility for adapting new algorithms and new data is important. For example, a framework can be used to support multiple models to avoid the work of rebuilding the system, and containers can be used to isolate different parts of the big system as described in subsubsection 4.3.1.1, making the system understandable for new members (subsubsection 4.3.1.4). However, as stated in subsection 2.3.1, the adaption process can be more challenging than in normal software engineering projects since the model customization requires data science expertise rarely found in pure software engineering teams. It also argues against the idea of short POCs mentioned in subsubsection 4.2.1.1 since developing a flexible system generally requires more resources than hard-coded solutions.

5.2.1.2 Working method – using Agile methodology

Connected to the challenge of developing systems with high modularity is the idea stated in subsection 4.2.2 that some researchers prefer not to work in traditional software engineering agile workflows. The mindset of software engineers, data scientists and researchers may differ in the sense that some researchers want to explore different models and approaches more thoroughly in order to present the best prediction possible. In each iteration, exploration and customization of new models take valuable time that is scarce in POCs, while software engineers have a tradition of working in agile environments and are more comfortable focusing on small value-bringing activities, better suited for short iterations. Another consequence of agile methodology in data science is that technical debt as described in subsection 2.2.1 may arise. One dept mentioned by Sculley et al. (2015) which also occurred in subsubsection 4.2.1.1 is the challenge of scalability. In contrast, there are also positive aspects with following an agile working method such as close customer contact that will be further elaborated in subsection 5.2.4.

5.2.1.3 Practical verification

From the practical verification, it can be concluded that high cohesion and low coupling, i.e. modularity, was important for the usability of the anomaly detection system. The success of the project was very much dependant on the fact that the two ML algorithms could be used separately from the system and the architecture provided an idea of how future algorithms could be tested. The project was also structured to follow an agile workflow which proved to be useful. For example, by keeping a close dialogue with Energy Machines, the number of sensors could be restricted to those who presumably would be affected by the potential failure, increasing the quality of the data. Through the project, it became clear that when working with limited resources and fixed deadlines, the room for extensive testing is small, but by following an agile methodology, we still managed to produce value for the client. Hence, even if there is a clash between agile workflow and high-quality predictions the former can still produce value within the time-scope.

5.2.2 Selection of techniques

The challenge of implementing an agile workflow in data science also concerns the selection of methods and algorithms. As suggested by Jarmul (2017) in subsection 2.2.3, most interviewees agree that they prefer working with simpler models as opposed to complex models. The reason is mainly that they are easier to understand, less prone to overfitting and robust. If the algorithm is hard to understand, it does not provide the user with information about how the output is produced, similar to a blackbox[2]. The risk of overfitting is a result of a lack of quality data and even if there is enough data, there will still be a need for large computational power. That said, it is clear from the results in subsubsection 4.1.3.2 that deep learning is a growing trend and that there are some areas where it shows great potential, for example in image recognition, video and NLP since these by nature contain compact information. Some interviewees also suggested the medical field as having sufficient data for deep learning. The main obstacle, in this case, is finding and signing the right forms in order to get access to public medical records. In the future, subsubsection 4.1.3.4 also suggest that GANS could be used for image and video augmentation which could help the training of autonomous vehicles.

5.2.2.1 Practical verification

It is not entirely fair to compare Isolation Forest and Dynamic Boltzmann Machine (DBM) since the later only target multivariate data, i.e clusters of sensors. However, both algorithms could predict the anomalies but in terms of usability, isolation forest was more straight forward since it only had one tuning parameter compared to five for DBM. Further strengthening the argument that simpler models may be preferable in short projects.

5.2.3 Matching Academia and Industry

One reoccurring problem expressed in subsubsection 4.3.1.2, not only in the ML field is that academia develops existing solutions but the applicability on real problems is hard to match, which is obvious with a technique such as GANS (described in subsubsection 4.1.3.4). One use case example being Snapchat's gender-swapping filter which might rely on the neural network CycleGAN model based on the billions of selfies as input. On the other hand, the industry is searching for solutions to problems that are hard to match. There is a gap between the two that can only be filled through further dialogue and cooperation. Another interesting result is that it is mostly the financial sector that has the economical prerequisites required for investing heavily in ML at the moment apart from large IT organizations. The entry cost for the ML project is too high for the average company. Introducing software engineers to support in the early ML activities is one way of bypassing the *lack of data scientists* bottleneck.

[2] Blackbox: You put something in and it spits something out and you do not know what happened or why.

5.2.3.1 Practical verification

The struggle to match academic and industry also became obvious in the early phase of the thesis. The stakeholders; Chalmers and IBM, did not integrate smoothly in their ideas about the project. It required three iterations before all stakeholders gave their approval of the initial thesis-proposal. Introducing Energy Machines to the project added another layer of complexity to the thesis since they also had their own idea for the project. However, after meeting face-to-face, discussing their needs and our resources, it became clear that we could integrate their main idea into the already existing scope.

5.2.4 Collaboration between stakeholders

This section covers the aspects of collaboration among different stakeholders and ML projects is largely dependent on a free flow of information. Transparency and bureaucracy are some important aspects that need to be addressed in most ML projects

5.2.4.1 Communication among experts and knowledge sharing

Another limiting factor related to the uniqueness of projects is the communication with domain expertise as described by López et al. (2018) in subsection 2.2.4 and expressed by interviewees in subsubsection 4.2.1.5 and subsubsection 4.2.1.6. The relevant features and the model results evaluation are in many cases in need of expertise input. The world is complex and one data scientist cannot be an expert in all fields and therefore needs support from people with relevant domain knowledge. One interviewee mentioned that this will lead to more specialized types of data scientists. Each field of science will get their own data analyst experts, for example, the bioinformatics field has transformed over the last ten years to include data science tools and methods in the research.

5.2.4.2 Transparency and explainability

Bureaucracy and confidentiality can also be restraining factors in this ML projects as described in subsection 4.4.2. The results suggest that the primary solution is an early quality check to determine data feasibility. If not, a list of suggested fixes should be presented to the client in order to start the project. It is also crucial to analyze the customer's needs and problem as ML might not be the best approach, to begin with. Furthermore, multiple interviewees mentioned that they no longer take on vague projects involving large data lake dumps as it is difficult and time-consuming to extract useful information as the result in subsection 4.4.2. Another solution is to keep a continuous dialogue with the customer throughout the project in order to specify the appropriate scope of the project and to deliver validated value to the client.

The gap in ML understanding between ML practitioners, managers and clients are also one of the themes mentioned in the result subsection 4.5.1. Focusing on simpler models is not necessarily in the best interest of a data scientist since it may not provide the best prediction. However, in many cases, the result of a simpler model can be seen as support for implementing more complex models, as well as providing better explainability for the client and high-level project managers. Some interviewees expressed that the interaction with people, such as visualizing and explaining the model, is the hardest part. If the model cannot be explained, uncertainty is introduced to the project, which could increase the risk of the project and should be avoided. From a data scientist consultant's perspective, it is necessary to explain the business value of the model result, by translating the ML output to understandable client language in common words based on client's technical knowledge level.

5.2.4.3 How transparency can help reduce the risks

The previous paragraph leads to the subject of risk, described in subsubsection 4.4.3.1. One risk is that ML solutions are "oversold", for example, if they can produce quick results. Most of the interviewees agree that it is the data scientist's responsibility to accurately convey the risk of their result to the best of their ability. Transparency should be stressed as stated by an interviewee. Furthermore, it is easy and dangerous to get blinded by the performance of models. For example, looking at numbers of true negatives may be of great economic value but in some cases such as in cancer prediction, the cost of sending a patient to a second referral needs to be weighted against the statistical uncertainty and the ethical aspect if the prediction is actually a false negative. Another example of two-layered risk mitigation is to restrict the ML models from making direct changes in the customer's environment, for example in financial markets, by not supporting direct stock trading but instead providing predictions to a trader who then makes the actual trade.

5.2.4.4 Practical verification

Bureaucracy and accessibility
As stated in section 4.6 the initial phase of getting access to IBM's system and Energy Machines data proved to be time-consuming. The external researcher has no real means of accelerating the process more than encouraging the stakeholders to do their part. It is understandable that companies cannot simply grant access to their resources without question but as junior data scientists it came as a surprise that the required time was more close to weeks rather than days, and that was even without extensive bureaucracy (no signed NDA). However, once the initial phase of

access was done the project flowed smoothly. In the later stages of the project, it also became more clear how irreplaceable the domain expert was for the outcome of the project. Since the anomalies found could not be validated, i.e. translated into faulty behaviour of the system without an expert from Energy Machines.

Explainability to the customer and team
In terms of delivery and explainability, the practical verification indicates that two junior software engineers can produce value for an industry partner even with limited previous ML experience. The most challenging part turned out not to be understanding and explaining the algorithms but instead assessing the required data structure and data quality. The explainability became a secondary focus until the data was in a usable format. Even if the explainability is deemed important by the interviewees, it loses in regards to data structure according to the practical verification. Throughout the project, we also actively conveyed the uncertainty of the project to the client and tried to mitigate the risk by following an agile strategy with a continuous dialogue with the client, domain experts and senior data scientists.

5.3 Thoughts about future ML trends

Some of the findings from the interviews had more of a futuristic character. In this section two of the more reoccurring ones are presented and analyzed as shown in Figure 5.4.

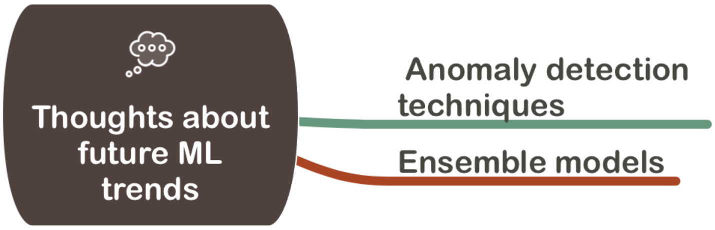


Figure 5.4: Mind map of Machine Learning trends

5.3.1 Ensemble models

An interesting question for the future is if ensemble models will become more widely used. In subsubsection 4.1.3.5 it is mentioned as a potential trend but the interviewees limited successful first-hand experience introduces uncertainty to the technique. It is believed to be best suited for projects which have multiple data angles. Hypothetically, this could be cases such as in medicine when you handle different types of data from the same patient. Another growing futuristic trend is the development of strong AI, which means the AI not restricted to one specific task as mentioned in subsubsection 4.1.3.3. Whether or not it will happen is uncertain but most interviewees predict that it will be developed eventually.

5.3.2 Anomaly detection techniques

The anomaly detection field has potential according to the interviewees in subsubsection 4.1.3.1. However, there are no straight answers to which algorithm that best fits a certain problem and data set. According to the interviews, they are no specific anomaly detection algorithms, instead, data scientists should explore and adapt different types of classification techniques which could, for example, be XGboost which is a ML library specialized in gradient boosting, Gated Recurrent Neural Networks (GRU) over Long Term Short-term Networks (LSTM). However, most interviewees express that linear models and tree models are preferable in most cases since they work on smaller data sets and are more forgiving in terms of data quality. It will be interesting to see if there emerge any specific algorithms in the anomaly detection field as it becomes more widely applied in industry.

5.4 Theory reflected by findings

Most of the technical challenges of ML presented in the discussion have already been addressed to some degree by other studies. For example, Sapp (2017) describe the initial ML activities and Sculley et al. (2015) describe technical dept related to ML. A new insight presented in this study is how software engineers can navigate between these challenges and still contribute to the ML project.

It is worth noting that Saleema Amershi (2019) reasoning surrounding their process maturity metric and identified challenges are to some extent reflected in this study. One example is that all levels of experienced people rank the challenge of data availability, collection, cleaning, and management as the hardest. Which was also implied by the interviewees in this case study as seen in subsubsection 4.2.1.3 but still, these activities are deemed possible for a software engineer to conduct. Saleema Amershi (2019) also describe three differences between developing ML applications and other IT application domains. One particularly interesting aspect being that the second differences that customizability and extensibility almost always requires a software engineer to have extensive data science knowledge are also reflected by the interviewees in this study, see subsection 5.2.4. However, the results of this study also argue that this knowledge can be obtained through close dialogue and information sharing with more experienced data scientists during a project and still be successful, i.e. it may not be required for a software engineer to have this knowledge prior to the project.

Reusability and platforms have been addressed in previous studies, e.g by Polyzotis et al. (2017) but capturing data science experience is a complex procedure and it will be interesting to see if ML becomes more accessible with presented techniques such as transfer ML and AI platforms. Will software engineers be able to easily use these techniques?

The conclusion that time and computational power affects ML projects is also already addressed by e.g. Sapp (2017) but the insight that it might lead to a preference of simpler models have not been identified in related theory.

The challenges with introducing a new member to a team were not deemed specific to ML projects. Instead, what was engaging was that no interviewee knew of any particular method to use in such situations. No theory was found beforehand related to the introduction of new members in ML projects. Maybe theory related to the introduction of the team member in software engineering teams can be applied to help solve the challenges in the future.

5.5 Threats to validity

The thesis result is mainly based on 16 interviews which are further tested through practical implementation, a POC. In this section, the validity of the thesis method and results are discussed as well as steps to mitigate potential threats to the validity.

5.5.1 Internal validity

Internal validity is the causal relationship between the conclusions presented in the discussion and the actual result. In this study, high validity means that the conclusions are correctly supported by interviews as well as through the practical test. If the interviews, the conclusions and the practical verification point in different directions, it may indicate a lower internal validity.

According to Runeson and Höst (2009) the validation of the result of a case study is related to the traceability of original answers and the interview format. Since all interviews were recorded and transcribed, all results should be traceable to at least one of the transcripts. However, due to confidentiality, the names of participants are not disclosed together with the citations. Runeson and Höst (2009) also mention that the demography of the participants and how different backgrounds of participants can give a wider information input. To tackle this, potential participants were contacted both inside and outside of Sweden, both men and women. Though the majority of the interviewees, in the end, had a strong connection to IBM and Sweden, we managed to conduct interviews with data scientists and researchers in Japan, USA, Ireland, and Switzerland. Some interviews were conducted with participants working outside of IBM and a third of the interviewees were women as described in Table 3.1. Furthermore, an emphasis was put in interviewing data scientist and researchers from different roles, junior, senior, polymaths, team leaders, etc as shown in Table 3.1.

Bias is a threat that the researcher should be aware of according to Runeson et al. (2012). In this case study, for example, bias was introduced in the discussion chapter chapter 5 since the practical verification is largely self-perceived. However, the practical verification helped support the results by putting to test some of the interviewees' statements. Not all results proved to be as "simple" or "hard" as implied by some of the interviewees but in general, the thoughts and meaning of the interviewees became clearer during the practical verification described in section 3.4 and section 4.6. For example, deciding the number of clusters may seem trivial to an experienced data scientist but not for a junior one. There is also the aspect of the interpretation of interviews and leading questions. Bias interpretations were avoided by recording all interviews as previously stated, making sure the information was kept solid.

In order to keep a transparent analysis, the main findings in the result chapter disclose the number of interviewees who support each theme. Some of the themes are more supported than others, for example, the results regarding data cleaning are among the most supported while potential future trends such as GANS are in the lower end. Naturally, future speculations regarding technology such as GANS were broad and each interviewee had unique thoughts. There is a risk that some of these less supported conclusions have a weaker correlation to the results since fewer participants talk about them. However, only in rare cases are the results supported by less than half of the interviews, instead, these findings where either deemed particularity interesting or emphasized by an interviewee.

5.5.2 External validity

External validity refers to the degree to which the findings can be generalized. If the results are based on a unique case it might be hard to test it in on a broader scale. Hence, lowering the relevance for other researchers, i.e. external validity.

According to Runeson et al. (2012) a case study is supposed to be useful for other application and studies with similar characteristics. Bridging the gap between applied ML in industry and academia is a broad term. However, the characteristics of the study are mainly based on interviews with people working for hands-on with researching and applying ML. There is no real niche except that a majority of the participants were working at IBM and that it can be challenging to get access to experienced data scientist. The possibility of replicating the study should be relatively high if the potential researchers could find interviewees with a similar background. The practical verification may be harder to replicate since the project in itself is unique in regards to data and the confidential aspects. The method of applying ML will need to be adapted to that specific case since data is a restraining factor.

There might also be a tendency towards certain results depending on the ratio between the interviewees' backgrounds as seen in subsection 5.2.1 where data science consultants verses researchers have a tendency towards different directions. Another interesting aspect would be to include software engineers with data science experience in future studies.

6

Conclusion

This thesis aimed to investigate the process of ML implementation in industry and intended to structure how software engineers, data scientists and researchers can work in order to progress in the field of applied ML. This was concertized by three research questions revolving around the implementation process as seen below. Data was collected through 16 interviews and further confirmed through a self-experienced practical implementation test.

6.1 Answers to research questions

RQ1:

> *What are the necessary tasks in the processes of embedding a ML algorithm in a software system?*

The process of embedding a ML algorithm in industry, in a narrow sense, consists of identifying the problem in the industrial scenario, assessing the scale of the problem, getting data access, understanding and preparing the data, iterated training with ML algorithms, validation of model output, in parallel with developing the surrounding software system (subsection 5.1.1 and subsection 2.3.1). Besides the purely technical work of data scientists and software engineers, the process in a broad sense, also consists of a managerial and communication aspect: interactions between humans, different stakeholders and different domain experts could be seen as more central than in general software engineering processes (subsection 5.2.4). In terms of complexity of the project and the acceptance of new ML techniques for the client, simpler models and communication that breaks down technical language can be seen as steps that increase the success rate of ML projects as presented in subsection 5.1.4.

RQ2:

> *What are the challenges in regards to new software environments, new data,*

> *new trends and new software engineers throughout the process of ML implementation in the industry?*

Challenges in the process of ML implementation in the industry can be seen from three different engineering roles as summarized in Table 5.1. From a data scientist's perspective, the most challenging part is the initial steps: understanding and collecting quality data, building a data pipeline from manual to automatic, choosing an effective way to clean the data. This is challenging because real-world problems and corresponding data varies greatly and is to a large degree uncontrollable. Another challenge for data scientist relates to the working methodology: the conflict between an agile-like methodology that focuses on quick consistently small value bringing tasks, and a research perspective where longer time sprints may be preferred in order to reach higher quality models.

From a software engineer's perspective, the challenge is more related to communication and cooperation with data scientists in ML projects. Apart from the common challenges in the software engineering area such as increasing the system modularity and working efficiently in an agile methodology. In order to bridge the knowledge gap, knowledge sharing through walk-throughs and dialogue with stakeholders is highly required. If this requirement is satisfied, a software engineer can in some cases contribute even in the more traditional data science activities such as data cleaning.

The challenge of communication and cooperation also exists among the whole team, managers and especially towards clients. Compared to common software engineering projects, the volatility of time scope and outcome for the ML project can be higher. As a result, conveying the uncertainty and keeping a relatively high level of transparency throughout the project mitigates the risk since it lets stakeholders make their own decision in regards to the model's accuracy, and potential impact.

RQ3:

> *How do time scope and increasingly rapid technological advancement affect ML practitioners decisions and strategy during the process of adapting ML algorithms, models and tools?*[1]

Cutting-edge ML techniques such as GANs and Deep Learning are currently being explored. Customers and data scientist might be tempted to use these hyped technologies but it might be safer to start with simpler techniques as described in subsection 5.1.4 and subsection 5.2.2. New algorithms and techniques emerge in large numbers in the wide-ranging ML area and some of them do not have extensive instructions, examples or documents, some are even impossible to replicate due to the fact that they require huge computational power, in many cases limited to large corporations. Especially for students and researchers in academia, the energy and

[1] Time, money, news and other external influences affect our everyday decisions. This research question aims to investigate how external parameters affect ML projects.

the access to resources such as computational power may be limited in comparison as discussed in subsubsection 5.1.2.1. From our result, most researchers will try to explore cutting-edge technologies and select the most optimal ones after considering the time, cost and match to their problem.

In projects involving paying customers, in order to increase the success rate, more and more data scientists are leaning towards simpler models, since the explainability of models and quick economical value for the client is prioritized. The timely, deliverable result through visualizations can quickly establish a client's trust and foundation for further projects. Although a solution from an early POC may not be perfectly suitable in a larger setting, some parts can usually be reused. In addition, extreme accuracy and cutting-edge algorithms may be tempting to use, but in practice, the industry may not need that level of accuracy and simpler algorithms can get good-enough results, sometimes ML may not even be the best approach to the industry problem, as mentioned in subsubsection 5.2.4.2. In order to increase the accuracy, multiple algorithms can be applied and ensemble models and transfer ML may be future answers to this as well as adding additional operational experts to strengthen the final decision. Adding a second layer of human interpretation may become increasingly important from an ethical perspective when implementing ML in e.g. health care, since the model will have a clear impact on peoples lives, as described in subsubsection 5.2.4.3.

In terms of tools, using platforms (see subsubsection 5.1.2.1) and frameworks is becoming increasingly more effective than building a project from scratch. Using cloud services for computational power and data storage may increase the scalability of projects. Furthermore, using libraries for algorithms and separating the system into containers, increasing the modularity could make the project more flexible and scalable as expressed in subsubsection 5.2.1.1.

6.2 Future work

This study revolves around the implementation process of ML, the challenges associated with it and how software engineers, data scientists, researchers and other stakeholders can structure their work to mitigate these challenges. As stated in the discussion, there is a need for further research regarding what degree simpler models increase or decrease the success rate of ML projects. Is it beneficial for the data science field as a whole or will it slow down the innovation rate?

Another interesting area for future research is how AI platforms can make ML technologies more accessible and how it will impact other industries. It would also be interesting to further investigate the ethical aspect of machine learning and AI, e.g. how people experience a computer-based prediction versus a medical doctor's. Is there a difference in trust and is it ethical to make systematical decisions based on a ML model's output. Furthermore, research regarding the prediction mentioned in the discussion: that all sciences will have specialized data scientists, would be interesting to look more closely at. Is it possible to be an expert in all types of data science or will the field progress to more separated fields of expertise.

References

Allen, R. (2018). *Business challenges with machine learning.* Medium. Retrieved 2019-02-06, from https://medium.com/machine-learning-in-practice/business-challenges-with-machine-learning-3d12a32dfd61

Arpteg, A., Brinne, B., Crnkovic-Friis, L., & Bosch, J. (2018). Software engineering challenges of deep learning. *CoRR*, *abs/1810.12034*. Retrieved from http://arxiv.org/abs/1810.12034

Bailey, J. (2008, 02). First steps in qualitative data analysis: transcribing. *Family Practice*, *25*(2), 127-131. Retrieved from https://doi.org/10.1093/fampra/cmn003 doi: 10.1093/fampra/cmn003

Bergstra, J., Yamins, D., & Cox, D. D. (2013). Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures.

Bergstra, J. S., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems 24* (pp. 2546–2554). Curran Associates, Inc. Retrieved from http://papers.nips.cc/paper/4443-algorithms-for-hyper-parameter-optimization.pdf

Bishop, C. M. (2006). *Pattern recognition and machine learning.* New York, NY : Springer, cop. 2006. Retrieved from http://proxy.lib.chalmers.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=cat06296a&AN=clc.b1350120&lang=sv&site=eds-live&scope=site

Black, J. (2019). *Cutting through the machine learning hype.* Forbes. Retrieved 2019-02-06, from https://www.forbes.com/sites/valleyvoices/2016/11/16/cutting-through-the-machine-learning-hype/#515c41755465

Cambride. (2019). *Cambridge advanced learner's dictionary thesaurus.* Cambridge University Press. Retrieved 2019-02-06, from https://dictionary.cambridge.org/dictionary/english/buzzword

Chandola, V., Banerjee, A., & Kumar, V. (2009, July). Anomaly detection: A survey. *ACM Comput. Surv.*, *41*(3), 15:1–15:58. Retrieved from http://doi.acm.org/10.1145/1541880.1541882 doi: 10.1145/1541880.1541882

Dominique Foray, . F. L. (2009). University research and public-private interaction. Retrieved from https://www.sciencedirect.com/science/article/pii/S0169721810010063

Foray, D. (2004). *Economics of knowledge.* MIT press.

Guion, L. A., Diehl, D. C., & McDonald, D. (2001). *Conducting an in-depth interview.* University of Florida Cooperative Extension Service, Institute of Food and . . . .

Jarmul, K. (2017). *The mummy effect: Bridging the gap between academia and industry (pydata keynote).* PyData. Retrieved 2019-03-29, from `https://www.youtube.com/watch?v=B3PtcF-6Dtc&index=9&list=PLGVZCDnMOq0oe0eD-edj_2CuBIZ938bWT`

Joblogic. (2019). *Introducing Anomaly Detection and Predictive Maintenance.* Retrieved 2019-01-13, from `https://www.joblogic.com/introducing-anomaly-detection-predictive-maintenance/`

Kim, M., Zimmermann, T., DeLine, R., & Begel, A. (2016). The emerging role of data scientists on software development teams. In *Proceedings of the 38th international conference on software engineering* (pp. 96–107). New York, NY, USA: ACM. Retrieved from `http://doi.acm.org/10.1145/2884781.2884783` doi: 10.1145/2884781.2884783

Lenberg, P., Feldt, R., Wallgren Tengberb, L. G., Tidefors, I., & Graziotin, D. (2017). Behavioral software engineering - guidelines for qualitative studies. *Journal of Systems and Software.*

López, B., Mordvanyuk, N., Gay, P., & Pla, A. (2018). Knowledge representation and machine learning on wearable sensor data: A study on gait monitoring. In *Proceedings of the first international conference on data science, e-learning and information systems* (pp. 45:1–45:2). New York, NY, USA: ACM. Retrieved from `http://doi.acm.org/10.1145/3279996.3280041` doi: 10.1145/3279996.3280041

Lwakatare, L. E., Raj, A., Bosch, J., Holmström Olsson, H., & Crnkovic, I. (2019). A taxonomy of software engineering challenges for machine learning systems: An empirical investigation. *XP 2019 (forthcoming).*

Menzies, T., & Rogers, M. (2015). *Sharing data and models in software engineering. [electronic resource].* Waltham, Massachusetts : Morgan Kaufmann, 2015. Retrieved from `http://proxy.lib.chalmers.se/login?url=http://search.ebscohost.com/login.aspx?direct=true&db=cat06296a&AN=clc.b2141016&lang=sv&site=eds-live&scope=site`

Polyzotis, N., Roy, S., Whang, S. E., & Zinkevich, M. (2017). Data management challenges in production machine learning. In *Proceedings of the 2017 acm international conference on management of data* (pp. 1723–1726).

Runeson, P., & Höst, M. (2009, April). Guidelines for conducting and reporting case study research in software engineering. *Empirical Softw. Engg.*, *14*(2), 131–164. Retrieved from `http://dx.doi.org/10.1007/s10664-008-9102-8` doi: 10.1007/s10664-008-9102-8

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

Russell, S., & Norvig, P. (2009). *Artificial intelligence: A modern approach* (3rd ed.). Upper Saddle River, NJ, USA: Prentice Hall Press.

Saleema Amershi, C. B. R. D. H. G. E. K. N. N. B. N. T. Z., Andrew Begel. (2019). Software engineering for machine learning: A case study. *XP 2019 (forthcoming).* Retrieved from https://www.microsoft.com/en-us/research/uploads/prod/2019/03/amershi-icse-2019_Software_Engineering_for_Machine_Learning.pdf

Sapp, C. E. (2017). Preparing and Architecting for Machine Learning. Retrieved from https://www.gartner.com/binaries/content/assets/events/keywords/catalyst/catus8/preparing_and_architecting_for_machine_learning.pdf

Sculley, D., Holt, G., Golovin, D., Davydov, E., Phillips, T., Ebner, D., … Dennison, D. (2015). Hidden technical debt in machine learning systems. In *Advances in neural information processing systems* (pp. 2503–2511).

Tensorflow. (2019). *An end-to-end open source machine learning platform.* Google. Retrieved 2019-05-34, from https://www.tensorflow.org/

Vaismoradi, M., Turunen, H., & Bondas, T. (2013). Content analysis and thematic analysis: Implications for conducting a qualitative descriptive study. *Nursing & health sciences*, *15*(3), 398–405.

Xing, E. P., Ho, Q., Dai, W., Kim, J. K., Wei, J., Lee, S., … Yu, Y. (2015). Petuum: A new platform for distributed machine learning on big data. *IEEE Transactions on Big Data*, *1*(2), 49–67.

Řehůřek, R. (2017). *The mummy effect: Bridging the gap between academia and industry (pydata keynote).* rare-technologies. Retrieved 2019-03-20, from https://rare-technologies.com/mummy-effect-bridging-gap-between-academia-industry/

A

Interview Questions

Background question

1. Can you tell us a little bit about your background, where you are from, education? (What is your position/roll right now?)

2. How long have you been working with ML? How long have you been working with ML anomaly detection, if any?

3. If you consider yourself a data scientist: what type of data scientist are you: Insight provider, Model specialist, Polymaths, Platform builder or Team leader (Kim et al., 2016).

    - Insight Providers, who work with engineers to collect the data needed to inform decisions that managers make.

    - Modeling Specialists, who use their machine learning expertise to build predictive models.

    - Platform Builders, who create data platforms, balancing both engineering and data analysis concerns.

    - Polymaths, who do all data science activities themselves.

    - Team Leaders, who run teams of data scientists and spread best practices.

Project Example

1. Can you walk us through a recent, finished ML project.

2. What was the most challenging part of that project? (which part takes the most time/money)?

3. What is the greatest insight in terms of working method/process you learned from that project, if anything?

Algorithm

1. What are your favorite ML algorithms/techniques, and why?

2. Are there any specific algorithms you think have potential in the anomaly detection field?

3. What are the pros and cons for these algorithms (what is the suitable area for this Algorithm)?

4. What are some recent trends in the general field of ML algorithms do you think?

5. Have you heard about the Dynamic Boltzmann Machines algorithm? it is a new neural network algorithm developed by IBM research center in Tokyo in 2016. It is different from many neural networks by having a memory capacity so that it can remember long term dependencies and behaviors of a time series.

6. Can you describe the process of implementing a fully trained ML model in an existing system or tool.

7. How do you package your model for easy implementation in a software service or a business use case?

New data

1. When starting a new ML project, what do you look for in terms of data?

2. What are common pitfalls when presented with new data in ML?

3. How do you select the features at the initial training model stage?

4. When using an external data source such as an API, producing real time data. What would be your initial steps in utilising that data?

5. How do you process the original data from the clients before training the model?

Models

1. To what extent do you need to retrain the models in regards to only changing smaller conjunction like settings or features?

2. How often do you retrain, what factors would you consider? (retraining time, cost, the model size)

3. Is there any specific situations where you reuse the finished models? (For example, A model trained with one company's data and reused for analyzing another company's data.)

Engineer

1. How do you most efficiently introduce and integrate a new software engineer into one of your ML projects.

2. Can you describe the latest introduction/integration process of a new member of your team.

Time and risk

1. Considering the time efficiency, how do you predict required time/effort for applying the new algorithm?

2. What happens if you can not deliver on that prediction/deadline? Do you trash the project or put in more money/time?

3. What do you consider to be a reasonable time scope for training a new model that you have no previous experience with.

4. If a much better algorithm/new tech is created during the process of your team's implementing, what will you do?(add more freedom general questions)

5. If the fully trained model malfunctions during an ongoing client operation. How do you consider the risk?

6. if the implementation is not a failure, but considering the commercial and market it's a failure, how do you deal with that?