NTNU
Norwegian University of
Science and Technology

Framework for Proactive Enterprises

Evaluation and Enhancement

Eirik Hamnvik

Master of Science in Computer Science
Submission date: June 2018
Supervisor: Sobah Petersen, IDI

Norwegian University of Science and Technology
Department of Computer Science

Framework for Proactive Enterprises - Evaluation and Enhancement

NTNU

TDT4900 Computer Technology
Master Thesis
Eirik Solberg Hamnvik
Project supervisor: Sobah Abbas Petersen
June 12, 2018
Norwegian University of Science and Technology

Abstract

Enterprises in today's world find themselves in fast-changing environments and should for that reason be applicable to change. By being proactive and agile, enterprises can be early adopters of newfound opportunities, threats or other future events and can react quickly[PvdKP16, MSB+14a]. In *A Framework For Helping Enterprises Position Themselves Towards Future Enterprises*[Ham17], which was a research project done during the fall of 2017 by the author of this thesis, a framework was developed to help enterprises position themselves and map their situation. By utilizing the framework, enterprises can get a better understanding of where they are and can compare their situation to the future enterprises defined by the EU commission[fin12]. The framework gives an enterprise an overview of strengths and weaknesses and in which areas they should improve. The framework has eight defined capabilities which are decentralization, interoperability, awareness, perceptivity, prediction, intelligence, action and extroversion.

The work done during this master thesis is enhancing that framework. The specific contribution this thesis does to the framework is to answer the question *how* an enterprise can improve within each of the different capabilities. The way this is done is by identifying approaches and technologies that can be used within each capability. It makes it easier for an enterprise to be able to start its transformation into being a proactive and future enterprise. The results of this thesis are mainly obtained through conducting a literature review and the development of a prototype.

The prototype is developed for a collaborating enterprise. Throughout this thesis, the collaborating enterprise will be referred to as *Business A*, due to anonymity. The prototype consists of the development of a prediction model to be used within predictive maintenance by Business A. The prototype is a contribution as it is included as a tutorial for enterprises. The prototype is used to enhance Business A illustrated through the framework. In addition, it enriches the framework by adding an example of an approach and technology.

The results of this thesis present an enhanced framework.

Keywords — Proactive enterprises, Framework for agile enterprises, Predictive maintenance

Acknowledgements

The completion of this thesis was done with help from several people. I am ever thankful for the help from my supervisor Sobah Abbas Petersen, associate professor at NTNU (Norwegian University of Science and Technology) and senior research scientist at SINTEF[1]. Her help and guidance has helped me throughout my thesis, and also during my research project. Petersen has helped me during my research in the form of being a domain expert and providing me with relevant articles where needed. Additionally, I would like to thank her for the help and guidance to apply for economic support from the department of computer science(IDI), and I wish to thank IDI for the acceptance of the economic support. The economic support covered the use of Amazon Web Services for storage of the enterprise data used, and also for the prediction model training done through there services.

I would like to thank the willing cooperation from Business A that provided me with data. Thanks to the data provided, I was able to develop a prototype for a prediction model which in this thesis was used to enhance Business A illustrated through the framework, and also provide a guide for how enterprises can start their journey towards being proactive. There are mainly three employees at Business A who I would like to thank for the time they have spent communicating with me and answering questions I had during the development. Since Business A will be held anonymous, the names of the employees are not included.

[1]https://www.sintef.no/en/

Disclaimer

In this disclaimer, I declare that this master thesis is my original work, and has not been submitted before to any institution or similar. The work is based on my research project done during the fall of 2017[Ham17]. I would also like to emphasize the fact that all sources used are acknowledged and that these are cited in the reference section.

Eirik Hamnvik 12.06.2018

Eirik Hamnvik Date

Contents

List of Figures

List of Tables

Part I
Introduction

This part will introduce the master thesis. It will give an introduction to the motivation for the task, the problem description, the scope and target audience and the outline of the report. During the problem description, there will also be a clarification of which university the master thesis was written for and which study program it was a part of.

1 Motivation

The business environment is changing rapidly and dynamically[MSB+14a]. There are new technologies that are being introduced, new approaches, new opportunities and law regulations which all need to be taken into account when an enterprise wants to move forward and stay relevant[LAK+15]. Taking advantage of new opportunities and technologies are essential to staying ahead of competitors and should be done at a fast pace[PvdKP16]. One way to keep up with changes is to be proactive. To be proactive means that an enterprise is prepared for changes and can do specific measures in advance of changes happening[NRZ14]. Being proactive makes it possible for them to adapt fast. A way to be proactive is to integrate ahead-of-time prediction[LP17] and intelligent decision-making systems[GBM00] into an enterprise. For enterprises, being proactive might not be clear and there is a need for a framework which can help them understand what it means and how it is possible to achieve it.

In *A Framework For Helping Enterprises Position Themselves Towards Future Enterprises*[Ham17], a framework was developed for enterprises of all kinds to use to evaluate themselves towards future enterprises. They would also get an idea of what should be their next steps to turn into a future enterprise, including being proactive. The framework, which is described more thoroughly in part II, helps enterprises increase their awareness of what is needed to be done. Evaluation of the framework was done during in *A Framework For Helping Enterprises Position Themselves Towards Future Enterprises* by testing it with two business cases. From here on, the framework will be referenced to as the *framework for proactive enterprises*.

The motivation for the master thesis came during the development of the *framework for proactive enterprises*. Increasing interest in proactive enterprises and how they can advance sparked the motivation for this master thesis. A newfound interest in machine learning also contributed to the motivation for the work done. There is no question that the future enterprises will need to be able to handle environmental change at a lot faster pace than today's enterprises. Collaboration opportunities are also factors that enterprises need to be able to welcome and integrate rapidly to succeed and stay relevant[NRZ14]. The proactive ability is therefore essential and needs attention.

2 Problem description and context

The master thesis at hand was written during the spring of 2018. It was written as a part of a five year integrated masters degree in Computer Science at the Norwegian University of Science and Technology. The department the program belongs to is called Department of Computer and Information Science (IDI).

There was no specific defined problem description for the master thesis, but project goals and research questions can be found in chapter 7. The thesis sparked from the *framework for proactive enterprises*, but could not have the same problem description as the research project.

The framework that was developed was not thoroughly tested and evaluated during the fall of 2017. For the framework to be useful to enterprises, there is a need to develop it further than what is already done. Enterprises have needs for staying proactive due to the fact of the rapidly changing environments they operate in[NRZ14], and for that reason, an enhancement of the framework is needed. The *framework for proactive enterprises* did not answer the question *how* an enterprise can advance as it did not provide approaches and technologies that could be used. The question of *how* will be answered in this thesis. The contribution of this master thesis is to enhance the *framework for proactive enterprises* by doing further research and evaluate it using a prototype developed.

The first half of the master thesis consisted of doing a structured literature review and opening communication with potential enterprises that could fit as a collaborating enterprise. The collaborating enterprise would provide relevant data that would be used to make a prototype of predictive maintenance. The second half of the master thesis consisted of developing the mentioned prototype for the enterprise. Work on the report for the thesis was done throughout the semester, and the report was frequently updated.

The prototype is developed using predictive maintenance. Predictive maintenance can be critical to enterprises as it can reduce costs of scheduled maintenance, increase the running time for enterprise machines and improve the overall efficiency of a system. Predictive maintenance makes sure that maintenance is done when it is genuinely needed[LP17, HB11]. Using predictive maintenance makes sure of more efficient and cost-effective use of enterprise resources[LP17]. Other advantages that are achieved by predictive maintenance are increased availability, quality along with increased productivity of equipment, while insurance and maintenance costs are decreased[eCCO12, AP17, IBM, SAP].

3 Scope and target audience

The master thesis shall be written and finalized during the spring semester of 2018. The deadline for the master thesis is the 20th of June, and this report with appendices will be the sole delivery to NTNU. A prototype will be developed and included in an appendix. The prototype and report will also be delivered to Business A. The master thesis will be finalized within 20+1 weeks after the thesis is started. The extra week comes from easter 2018. The master thesis is written solely by Eirik Hamnvik, but during the project, the project supervisor will have access to the report and may come with advice. Aids and contribution can be given by other people as well, but will then be credited.

The target audience for the master thesis is enterprises of all sizes and especially enterprise leaders or managers who want to change their enterprise and make it proactive. Using the framework for proactive enterprises can be done by any enterprise no matter the size or budget. The prototype in this thesis targets a smaller group of enterprises, which generates a lot of data and that have something to win on introducing predictive maintenance in their enterprise.

The framework helps enterprises turn proactive, and a short description of proactive enterprises are given. Since enterprises are facing an increased amount of pressure towards their products[BGB, PvdKP16], a new form of enterprises has been developed. These enterprise types are called agile enterprises, proactive enterprises or proactive sensing enterprises[PvdKP16, NRZ14, LAK+15, MSB+14a]. As mentioned, the pressure towards high quality and low prices have forced enterprises to look for opportunities to save expenses in every possible corner[BGB, PvdKP16]. This pressure is what resulted in the forming of the agile enterprises, as the enterprises needed to maintain quality while reducing prices. Proactive enterprises are enterprises that predicts future scenarios by utilizing intelligent systems and therefore can act ahead of something happening.

Several factors were a part of the decision for choosing the collaborating enterprise. The author looked for an enterprise that was generating a lot of data, which had the opportunity to share the data, and that can gain on predictive maintenance. The enterprise that ended up being the collaborating enterprise is generating a lot of data as they have a vast amount of enterprise machines that are in daily use. Through contact with the enterprise, it was possible to determine that they were able to share the data with the author. Furthermore, as later sections will show, there was an opportunity for the enterprise to take advantage of a prototype such as the one to be developed.
Another factor that contributed to the decision was that the enterprise chosen was easily reachable by the author, and personal relations made initiating contact more comfortable.

4 Report outline

This master thesis consists of ten parts. These parts include the introduction, background, research methodology, preliminary study, emerging and relevant framework technologies, prototype development, results, discussion, conclusion and future work, and appendices. **Part I** is giving the reader an introduction to the rest of the thesis.

Part II gives an explanation and introduction of the *framework for proactive enterprises*. This part is included to give the reader a presentation and better understanding of the foundation that was set before starting this thesis. **Part III** gives a presentation of how work was done while writing this thesis by presenting the research methods used.

Part IV introduces some concepts that were researched during the research project and prior to the start of the master thesis. It is relevant for this context but were not relevant for part II. **Part V** presents the results of the literature review done of emerging and relevant technologies in connection to the framework described in part II.

Part VI will describe the development and testing of a prototype developed in collaboration with Business A to enhance the enterprise, illustrated through the framework.

Part VII will present the results from the development of the prototype and the results of the enhanced framework described in part II. Discussion of the thesis and the results will be done in **part VIII**. **Part IX** will conclude the thesis and present future work that can be done to enhance the framework further.

In **part X**, the appendices are included.

Part II
Background

This part of the master thesis gives an explanation and presentation of the *framework for proactive enterprises* that was developed during the fall of 2017[Ham17]. The development process was a full semester project which was the preparing work for this master thesis.

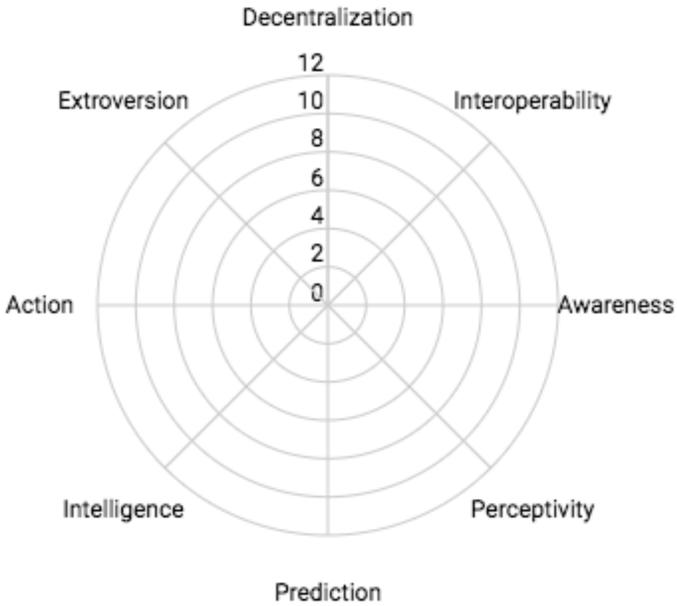
5 Explanation

During the research project, development of a framework which can help enterprises develop further towards a set of defined future enterprises was done. The work on the framework is the background of this master thesis. During the development process, evaluation of an enterprise, Business A, was done by utilizing the framework, and the enterprise was classified as a liquid enterprise as defined by the EU Commission[fin12]. This classification can be found in chapter 22, where it can be seen that Business A was classified between being a *liquid enterprise* and a *partially proactive enterprise*. A liquid enterprise "is an enterprise having fuzzy boundaries, in terms of human resources, markets, products and processes. Its strategies and operational models will make it difficult to distinguish the 'inside' and the 'outside'"[fin12]. A partially proactive enterprise is an enterprise that has introduced proactive behavior in some or all of their systems but that is not proactive as an enterprise in total. Business A was lacking some capabilities, including proactivity, intelligence, and the sensing capability, as defined in the framework. This sparked an interest for developing a program that could help them advance as an enterprise, and make them more proactive. Decisions were made to help them develop a system for predictive maintenance where machine learning would be used to predict if one or more of their enterprise machines were likely to fail within a short period. If accomplished and integrated into their systems, the prediction model would help them advance as an enterprise and expand there results within the framework. Chapter 6 will give a brief description of the framework and how it should be used.

6 Framework for proactive enterprises

NOTE! This section was also included in the research project report.

The framework consists of a radar chart that has eight different pillars specifying a capability each. The different capabilities are decentralization, interoperability, awareness, perceptivity, prediction, intelligence, action and extroversion. Eight tables, one for each capability, are presented alongside the framework. They are used to create the graph an enterprise will end up with when using the framework.
For each capability in the framework, an enterprise should look at the belonging table and find the value based on the characteristic that describes the enterprise best. For an enterprise to reach a value based on a characteristic, they need to be described by the characteristics prior to that one as well.
After a graph has been drawn up, it is possible for enterprises to see which strengths and weaknesses they have, and therefore move focus to capabilities they want to improve.
As the framework consists of eight capabilities, combinations of the different capabilities make strengths areas that can be identified. Strength areas can be seen in the combination box beneath the radar chart. By fulfilling a good score in a combination, an enterprise can be classified as having a strength within the combination capability specified. By not fulfilling the combination, it will be classified as a weakness. The combination of extroversion, decentralization, and interoperability together make the ability of collaboration and shows that an enterprise is collaborative in different settings. Decentralization, interoperability, and awareness together make the ability to change fast and be agile. The sensing capability is a combination of awareness, perceptivity, and prediction. To describe an enterprise as intelligent, a good score within the combination of perceptivity, prediction, and intelligence is needed. The proactive capability is a combination of prediction, intelligence, and action. The combination of awareness and perceptivity can classify the enterprise as aware.



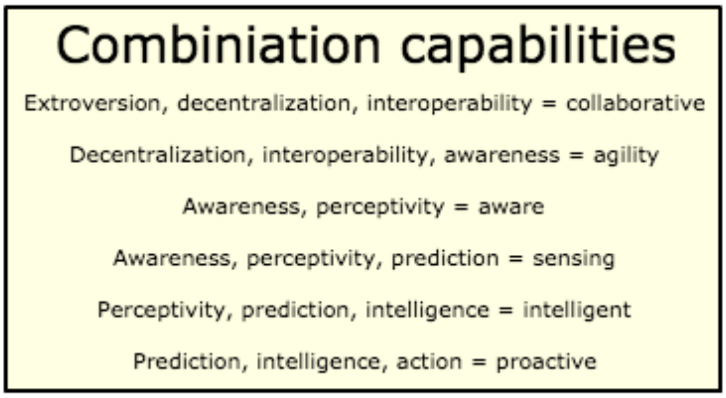


Figure 1: The framework

Specification	**Description**	**Value**
Centralized	The enterprise is centralized and works with the top-down approach. Decisions might have to go through several layers of management which creates delay	0
Decentralized intelligence in systems	Systems consists of separate parts which are able to make own decisions	4
Self-organizing teams	Self-organizing teams work for their own best and decisions can be made rapidly	8
Emergent knowledge spaces	Teams share information and learn from each other which secures collaboration from inside the company	12

Table 1: Decentralization

Specification	Description	Value
No interoperability	No exchange of data.	0
Data interoperability	Basic exchange of analog and digital data	3
Service interoperability	Enterprise system able to collaborate on a functional level	6
Process interoperability	Systems are able to do coordinated service exchange	9
Business interoperability	Multiple enterprise systems are capable of acting together	12

Table 2: Interoperability

Specification	Description	Value
No awareness	The enterprise has no awareness at all.	0
Internal system awareness	Systems are aware of conditions and status of different parts within themselves	4
Internal enterprise awareness	Internal awareness can include organizational structure, learning capability and technical infrastructure among more	8
Environmental awareness	Environmental awareness can include position in market, government support and new business opportunities among more	12

Table 3: Awareness

Specification	Description	Value
Not perceptive	The enterprise does not gather any data/information	0
Historical data and knowledge domain	Handling of historical data and the knowledge domain	6
Real-time	Real-time handling of data provided by hard and soft sensors	12

Table 4: Perceptivity

Specification	Description	Value
No prediction	There is no prediction done	0
Ahead-of-time prediction	The enterprise is anticipating the future based on ahead-of-time processing based on analysis of real-time, historical data and knowledge domain	12

Table 5: Prediction

Specification	**Description**	**Value**
No intelligence	The enterprise has no intelligence	0
Assertion, storing and acquisition of behavior patterns	Behavior patterns are discovered and stored which can impact decisions	4
Automated decision making	Machine learning processes makes decisions based on business goals and situational awareness	8
Continuous learning	Continuous improvement of decision making through sensor-enabled feedback	12

Table 6: Intelligence

Specification	**Description**	**Value**
No automation	Actions are taken reactively	0
Proactive Human based	Actions are taken proactively by humans	6
Proactive automated	Actions are taken proactive, rapidly and automated by machines or processes	12

Table 7: Action

Specification	**Description**	**Value**
Closed innovation	The enterprise is based on closed innovation and knowledge ownership, not willing to share information and experience	0
Blurry boundaries	Introduces communication with external sources. Can gain trust outside of enterprise and secure relationships	4
Sharing and usage of experience and knowledge	Sharing information to embrace innovations and collaboration processes	8
Integration and coordination between partners	A high degree of integration between partners to secure alignment of individual plans to achieve a joint goal	12

Table 8: Extroversion

Part III
Research methodology

This part of the thesis will present the project goals and the research questions to be considered. A clarification will be made of what the expected outcome of the thesis is. Furthermore, the research methods used in this thesis and study will be presented.

7 Project goals and research questions

This chapter will present the primary goal of the master thesis, but also the different research questions that are taken into consideration during the work.

7.1 Project goals

Throughout the research project during the fall of 2017, development of the *framework for proactive enterprises* was done. In this master thesis, the framework will be strengthened by identifying relevant technologies to support enterprises and by developing a prototype. This prototype will be used as a source of experience, which can be useful for enterprises that want to try similar development. The prototype development and experiences will be presented in this thesis.
This work will review and obtain an overview of the relevant technologies that are currently available to help enterprises within the different capabilities defined in the framework. The results of this review will include recommendations on how and when the relevant technologies could be used. The work will also include the implementation of a prototype using one of these technologies; more specifically Machine Learning. The work will be based on a real case on predictive maintenance of enterprise machines in an enterprise.
The project goals are as following:

- Conduct a literature review of emerging and relevant technologies
- Enhance the *framework for proactive enterprises*
- Identify a relevant technology and methodology to develop a prototype
- Develop and evaluate the prototype

The expected outcomes of this project will be an enhanced framework for describing proactive enterprises and how technologies support these capabilities. The enhanced framework will consist of not only the initial framework, but also a table which will define approaches and technologies that can be used for each of the capabilities within the framework. Besides, an example of how technology could help this will be demonstrated.

The overall and final project goal can be defined as such:

The project goal is to enhance the "framework for proactive enterprises"

7.2 Research questions

The research done in this master thesis is based on the following defined research questions.

- RQ1: Which technologies exists for supporting the different capabilities within the *framework for proactive enterprises*?
- RQ2: Which approaches are used for the different capabilities?
- RQ3: Can results found by addressing RQ1 and RQ2 enhance the *framework for proactive enterprises*?
- RQ4: How can developing a prototype for predictive maintenance using machine learning help enhance the *framework for proactive enterprises*?
- RQ5: How is it possible to create value from existing data to make enterprises more proactive?

8 Research method

This chapter will present the different research methods used while doing research. A structured literature review and development and evaluation of a prototype were done.

8.1 Literature review

For finding relevant and emerging technologies that could support the different capabilities in the *framework for proactive enterprises*, a *systematic literature review* was done. As presented in this section, some of the research strategies used are *domain expert contact* and *keyword search*.

Anders Kofod-Petersen has written a technical report on how to do a structured literature review in computer science[KP12]. His ideas and guidelines are the base of the structured literature review strategies used in this thesis.

8.1.1 Planning

Identification of the need for a review. The need for the literature was identified during a discussion of the master thesis with the project supervisor. During the discussions, it was specified further what the thesis would be about and which topics needed literature review.

Specifying the research question(s). After discussions with the project supervisor and after thorough consideration the research questions were specified. The research questions are described and presented in section 7.2.

Developing a review protocol. A review protocol was developed which was the template for how the literature review should be done. Here a presentation of the review protocol is done so the work can be reproduced.

1. Select potential relevant articles from the internet through keyword search through Google Scholar and Oria[2]
2. Categorize articles by relevancy and field of use by reading abstract
3. Relevant articles included for further evaluation with full text review
4. Discard articles that are not useful or relevant enough
5. Include useful information from relevant articles in document
6. Include most useful information from the document into literature review
7. Cite sources

Evaluating the review protocol. Evaluation of the review protocol was done by iteratively using the protocol while conducting the literature review.

8.1.2 Conducting

By utilizing the developed review protocol from the planning step, it is possible to conduct the actual literature review.

Identification of research. This step is designed to help retrieve all relevant literature for the defined research questions. Mainly two search strategies were used to accomplish this. The first strategy used *domain expert contact.* Some relevant articles were given to the author by the project supervisor during the research project done during the fall of 2017. Some of these were relevant also for the master thesis. Another strategy utilized was *keyword search.* Mainly Google Scholar and Oria were used during searching as they give access to an extreme amount of sources and articles. By grouping keywords into at least three groups, the author was able to find a reduced set of relevant articles for each topic.
The author will present some of the different groups used for finding relevant literature.

[2] literature search tool at NTNU

	Group 1	**Group 2**	**Group 3**
Keyword 1	Decentralized system	Communication network	Decision making hierarchy
Keyword 2	Approach	Technology	Connected decision-makers
Keyword 3	Information System	Information technology	

Table 9: Decentralization keyword search table

	Group 1	**Group 2**	**Group 3**
Keyword 1	Communication protocols	Machine-to-machine	Business-wide
Keyword 2	Interoperability	Systems	Integration
Keyword 3	Technology	Interoperability	Technology
Keyword 4	Ontology	Semantics	System

Table 10: Interoperability keyword search table

	Group 1	**Group 2**	**Group 3**
Keyword 1	enterprise	enterprise	risk management
Keyword 2	internal	external	swot
Keyword 3	environment	awareness	soar
Keyword 4	approach		

Table 11: Awareness keyword search table

	Group 1	**Group 2**	**Group 3**
Keyword 1	Big data	Data handling	real-time
Keyword 2	management	real-time	analysis
Keyword 3	techniques	technologies	data

Table 12: Perceptivity keyword search table

	Group 1	**Group 2**	**Group 3**
Keyword 1	Prediction	Prediction	Oracle CEP
Keyword 2	Complex Event Processing	Machine learning	Streambase
Keyword 3	Real-time	Amazon Sagemaker	Technologies
Keyword 4	Pattern matching		

Table 13: Prediction keyword search table

	Group 1	**Group 2**	**Group 3**
Keyword 1	Intelligent	Intelligent systems	Decision support
Keyword 2	Decision support system	Decision making	Web-driven
Keyword 3	Data-driven		Framework
Keyword 4	Model-driven		

Table 14: Intelligence keyword search table

	Group 1	**Group 2**	**Group 3**
Keyword 1	Automated processes	Human-based	Actions
Keyword 2	Action	Actions	Feedback
Keyword 3	Proactive	Reactive	
Keyword 4	Predictive		

Table 15: Action keyword search table

	Group 1	**Group 2**	**Group 3**
Keyword 1	Enterprise	Collaboration	Enterprise
Keyword 2	Collaboration	Networks	Open
Keyword 3		Cross-boundaries	Boundaries
Keyword 4			Cooperation

Table 16: Extroversion keyword search table

Selection of primary studies. The previous step gave the author a far too big selection of articles and relevant literature that he needed to evaluate. The author needed to reduce the set of articles, and some strategies were used. Where there were too many articles about a certain technology, the author would normally look for relevant articles written since 2014. Filtering by year reduced the set enough to filter by quality.

Study quality assessment. To filter away and further reduce the set of relevant literature some criteria were defined. Table 17 define some inclusion criteria (IC) and quality criteria (QC) for the literature review. Some articles got through the criteria equally, like internet articles without abstracts and similar.

Criteria identification	Criteria
IC1	The study is mainly about technologies relevant to the thesis
IC2	The study focuses on one of the aspects of one of the technologies relevant to the thesis
IC3	The study can somehow be connected to the research questions
QC1	The abstract gives a clear statement of what the study is about
QC2	The abstract gives a clear statement of the results of the study

Table 17: Inclusion and quality criteria for literature review

Data extraction and monitoring. Data extraction was done by reading the final set of relevant articles and gathering important data from them in different documents categorized by technology or field they were about. All articles that were relevant to each other were connected to each other in the same documents.

Data synthesis. Data synthesis was done by going through all the data gathered in the data extraction step. The data was used to give the author a good impression of the data and to be able to use the data relevantly. All data extracted and used in the report is cited.

8.2 Prototype

This section will present which development method was used during creation and development of the prototype for Business A. The development method chosen was *scrum*[Sch97], which is an *agile methodology*. For contact with Business A, the author chose *structured interviews* as the methodology.

8.2.1 Enterprise contact and structured interviews

Several factors were a part of the decision for choosing the collaborating enterprise. Requirements for an enterprise to qualify as a potential collaboration partner were that the enterprise generated a lot of data, that the enterprise had the opportunity to share the generated data, and that the enterprise could gain on introducing predictive maintenance. These were the requirements that lead to the collaboration with the enterprise chosen.
Another factor that contributed to the decision was that the Business A was easily reachable by the author, and personal relations made initiating contact more comfortable.

In December 2017 the author made initial contact with Business A which ended up being the collaborating enterprise. Interest was expressed in their machines and a short introduction of the project was done. Initial unstructured talks were held where we discussed the possibility for a collaboration where they would provide the author with data and the author would develop a system for predictive maintenance based on this data. No further contact was made until after Christmas and in January.

After meeting the thesis supervisor, contact with Business A was again made. A structured interview was done with one of their employees. The questions included in the interview is presented in appendix B. The interviewee could not answer all of the questions at the time. Some of the questions were answered, and by that, the author could know that the type of data that would be received was status reports from their enterprise machines.

During the timespan January to March there were ongoing talks about how this would be done. The author needed to wait for the data for quite a while as the data and status reports had to be made anonymous before receiving them. During the beginning of March, they were ready to transfer the data and make it available to the author. Another structured interview was done, this time with another employee at Business A. The form used for the interview can be found in appendix C. Decisions were made that the best way to transfer the data was by them uploading it to an AWS S3 bucket[3]. He told that the author could receive status reports that were as old as 8-9 months from 35-40000 different machines located in several countries in the world. After the interview a user profile were made for the employee at Amazon Web Services to give him access to upload the data to an S3 bucket created. Credentials were sent to him.

To be able to use the data received from Business A, there was a need to sign two legal documents. Firstly, the author needed to sign an agreement assigning the rights to the results of the project to Business A. Secondly, the author had to sign a non-disclosure agreement for the data received. Both documents were signed and sent the 02.03.2018.

Different data collection methods can challenge and raise concerns about validity and reliability, but in this case, this should not be a problem as the data is generated and delivered by a well-known enterprise which has nothing to gain on altering the data received.

8.2.2 Prototype development

For the prototype development, agile development was used, and more specifically the scrum framework[Sch97]. To track the tasks that needed to be done, a scrum task board was set up

[3]https://aws.amazon.com/s3/

which included different categories. The categories were the *backlog*, *under development*, *testing* and *done*.
Tasks were defined while developing the necessary programs to handle the data. Concerning the fact that the author did the development by himself, no traditional scrum meetings were held, but sprint planning was done solely by the author. Every week of the development period, goals were set for the coming week and what should have been accomplished by the end of the week.

Figure 2 shows the typical workflow for scrum. Tasks are inserted into the backlog, and certain tasks are chosen to be included in each sprint. This is how development was done for the prototype.

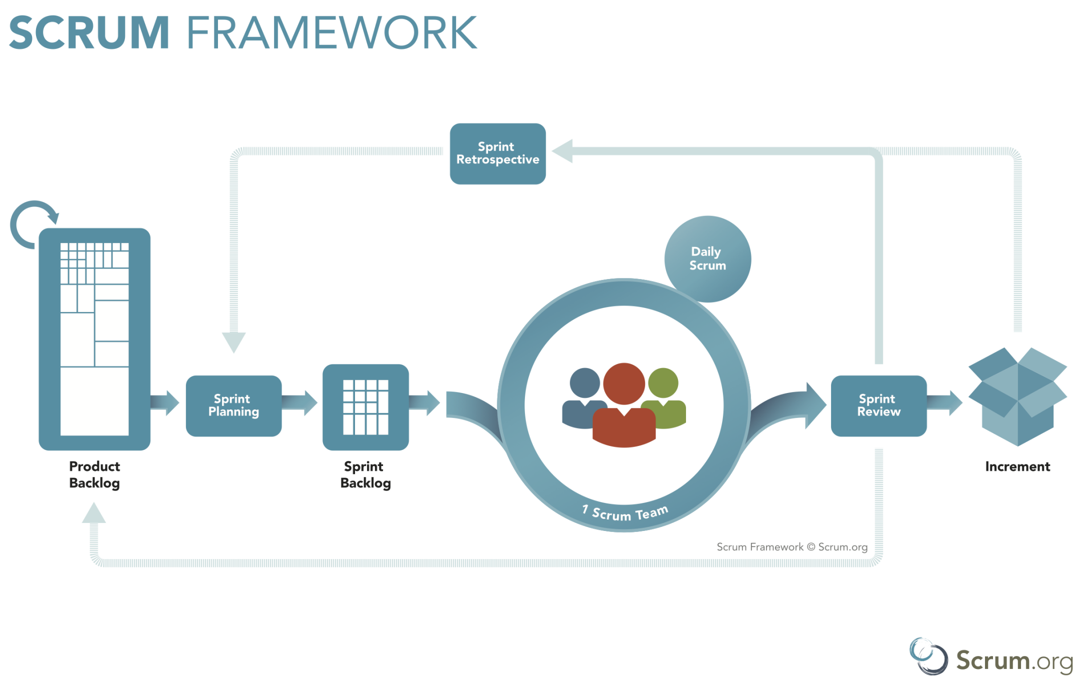


Figure 2: The scrum framework[Scr18]

8.2.3 Prototype evaluation

Evaluation of the prototype will be done by evaluating the predictive model generated. It is essential to evaluate the model to see whether it will be able to do a reliable job of prediction with new data. Since the new data which will be run through the model in the future does not have a target column specifying the actual value, there is a need to check accuracy for the prediction model. This can be done by splitting the used dataset into two sets, where one of the sets holds training data and the other set holds test data. The data will be split as such; 70% of the data will be contained in the training dataset, and the remaining 30% will go into the test dataset.

Testing will be done after training is complete, where data from the test set will be sent to an endpoint created. Then a comparison can be made to the actual values, which are known, against the values that the prediction model returned.

The author will look at four metrics of evaluation of the prediction model. The four metrics are accuracy, precision, recall and the F1 score as presented by Amazon Web Services[Ser18b]. In the formulas, the abbreviations TP, TN, FP, and FN are used, and these are explained in table 18.

Abbreviation	Explanation
TP	True positive - Running machines correctly identified as running machines
TN	True negative - Not running machines correctly identified as not running machines
FP	False positive - Not running machines incorrectly identified as running machines
FN	False negative - Running machines incorrectly identified as not running machines

Table 18: Prediction model evaluation metrics abbreviations explained

Accuracy
Accuracy calculates the fraction of the predictions that was done correctly. The value of accuracy can be between 0 and 1, and a higher value indicates a better prediction accuracy. Accuracy has the following formula:

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN}$$

Precision
Precision calculates the fractions of correct values out of all values that predicted a running machine (in our prototype). The value of precision can be between 0 and 1, and a higher value indicates a better prediction accuracy. Precision has the following formula:

$$Precision = \frac{TP}{TP + FP}$$

Recall
Recall calculates the fraction of the correctly predicted running machines out of all the machines that were actually running. The value of recall can be between 0 and 1, and a higher value indicates a better prediction accuracy. Recall has the following formula:

$$Recall = \frac{TP}{TP + FN}$$

F1 score
The F1 score calculates the harmonic average of the precision and recall of a prediction model. The value of the F1 score can be between 0 and 1, and a higher value indicates a better prediction accuracy. F1 has the following formula:

$$F1 = \frac{2 * precision * recall}{precision + recall}$$

Part IV
Preliminary study

This part presents some concepts that were studied during the development of the *framework for proactive enterprises*. These are included and put in this part as they are important to the project done, but are not an answer to some of the research questions. Proactive enterprises are one of the topics included where a definition will be presented, what they are and which characteristics they have. Predictive maintenance is also included as it is what the prototype development is about. A definition, a presentation of what it is, areas of use, reasons for using it and other types of maintenance will be presented. The basics of Machine learning is also discussed. Each topic has a summary part to conclude it.

9 Proactive enterprises

This chapter will present the concepts of proactive enterprises and describe what they are and some of their characteristics. This concept is included as it is an essential concept for this thesis and because it is the foundation for the process that lead to the work done in this thesis.

9.1 Definition

A proactive enterprise uses intelligent systems to predict future scenarios and use this information to behave accordingly.

9.2 What are proactive enterprises

Enterprises are facing increased pressure towards production and quality due to the increasing competition and globalization within the markets[BGB, PvdKP16]. The pressure has led to a form of enterprises that not only needs to be able to sense both external and internal factors and environments, but that is also required to respond rapidly to changes within these. Enterprises of this type are called agile enterprises, proactive enterprises or proactive sensing enterprises[PvdKP16, NRZ14, LAK+15, MSB+14a]. As mentioned, the pressure towards high quality and low prices have forced enterprises to look for opportunities to save expenses in every possible corner[BGB, PvdKP16]. This pressure is what resulted in the forming of the agile enterprises, as the enterprises needed to maintain quality while reducing prices.

The environments that enterprises find themselves within are in constant change[BGB]. On a daily basis, enterprises end up in situations where it is necessary to make decisions based on their current situation[BGB, PvdKP16]. An example of an external environmental change can be newly opened governmental support[PJH16]. Furthermore, new technologies might emerge, new laws might be granted, or other restrictions or opportunities might surface. For an enterprise to be able to compete, it has to change rapidly to these restrictions and opportunities.

Proactive enterprises can continuously adapt and reorganize themselves[WMC+16]. Given the internal and external changes they monitor, they can reorganize and restructure quickly. They might have to add another department or change business goals. When an enterprise is considering change, some factors are important to consider. Their company and production goals and external factors such as customer relations are fundamental and need to be thought through when an enterprise is deciding the direction to go[BGB].

Proactive enterprises take advantage of sensor technology to gather an enormous amount of data[BPM+16]. One of the fields where an enterprise can behave proactively is maintenance. Traditional maintenance was done reactively, while the way to do it now is predictive. Predictive maintenance is anticipating future faults and errors on machines and equipment and acting before the actual fault or error occur[PvdKP16]. In chapter 10, we will look closer at predictive maintenance and characteristics, where we will see that predictive maintenance can minimize unexpected maintenance costs[PvdKP16].

Being proactive gives enterprises several advantages and opportunities[MSB+14a]. The ability to act ahead of time will give enterprises the possibility to take advantage of new business opportunities before the competitors can act on them. The enterprises will also have the ability to respond faster to internal and external environmental changes.

There are several challenges that face proactive enterprises[LAK+15, MSB+14a, WMC+16]. Among these challenges are context awareness, dynamic configuration, information requirement and processing and scaling. To be able to handle all the data gathered is a concern given the amount of sensor data enterprises can collect. Systems must handle the incoming data at near real-time, and this requires smart systems that can make intelligent decisions based on the enormous amount of data that enterprises have available[WMC+16].

9.3 Characteristics

In the *framework for proactive enterprises*, eight characteristics a proactive enterprise needs to have were defined. These characteristics are decentralization, interoperability, awareness, perceptivity, prediction, intelligence, action and extroversion[Ham17].

- *Decentralization* enables innovation, fast decisions and enables the opportunity to change rapidly.
- *Interoperability* enables innovation and sparks collaboration processes.
- *Awareness* includes both internal and external awareness and are important to understand what is going on both inside and outside of the enterprise. It is crucial to be aware of new business opportunities.
- *Perceptivity* is about making meaning of the data generated both within and outside the enterprise.
- *Prediction* is done by utilizing multi-dimensional information and can help enterprises anticipate the future.
- *Intelligence* is about making the right decisions based on the predictions an enterprise can make.
- *Action* is how the enterprise acts on the decisions made by the enterprise. Automated actions acts faster than humans for example.
- *Extroversion* is about how good an enterprise is concerning collaboration with other enterprises to finalize a product together.

9.4 Summary

This chapter has presented proactive enterprises and what they are. It is stated that there is a need for enterprises to anticipate the future and to be able to act upon the future scenarios they predict. Given the *framework for proactive enterprises*, there are several steps and enterprise type levels an enterprise can be at before becoming entirely proactive. Predictive maintenance is one of the concepts where enterprises can start in their journey of becoming a proactive enterprise. A subgoal of this thesis (RQ4) is to develop a predictive maintenance system for an enterprise to help them advance as an enterprise.

There are lines that can be drawn between predictive maintenance and proactivity. Introducing predictive maintenance can help an enterprise become more proactive by taking advantage of the capability of foreseeing future scenarios and then act on them before it actually happens. As will be seen in the following chapter, there are several advantages for an enterprise by introducing predictive maintenance.

10 Predictive maintenance

This chapter presents the concepts of predictive maintenance as well as a brief look at other types of maintenance. Some advantages and disadvantages of predictive maintenance are discussed.

10.1 Definition

Predictive maintenance is monitoring of systems and components within them and utilizing history and real-time data to predict when one of the components or systems might fail.

10.2 What is predictive maintenance

Industrial equipment and machines are getting more and more complicated, and require more thought through maintenance[VHP13]. Cost-effective and efficient maintenance is important and should be done by utilizing predictive methods. The primary goal of predictive maintenance is to be able to foresee when failures within equipment might occur and then an enterprise can be able to take appropriate measures[AP17, Fii18, Too16]. Predictive maintenance is done by the use of sensors to continually monitor the states of the different components in a system and the system as a whole[LP17, Maz16].

10.3 Areas of use

Predictive maintenance can be used for many different reasons such as failure prediction, failure diagnosis, failure detection, maintenance actions or failure type classification among others[fM15].

Predictive maintenance is typically used for predicting the remaining useful life of components or whole systems[VHP13]. By doing this, it is easier to schedule maintenance at the correct time, and also being able to combine maintenance of different components at the same time, without risk of replacing a well-functioning component.

An example is discussed in the article "*Improving rail network velocity: A machine learning approach to predictive maintenance*"[LPH+14]. The authors used mechanical condition detectors to monitor temperature, strain, infrared, vision, impact and weight among other things to predict future failures, so they could avoid service interruptions along with increasing rail network velocity.

10.4 Reasons for using it

Being able to schedule maintenance at the right place and time is very important for system engineers[LP17]. Adding the fact that it should be as cost-effective as possible makes it even harder.

Preventive maintenance is still very widespread amongst enterprises, but as mentioned, it is hard to schedule, and above all, it can end up being costly by doing unnecessary maintenance[LP17, Tri16]. It ends up with resource wastage[AP17]. While doing maintenance, machines have to be taken out of operation, engineers or other maintenance specialists need to focus their work on the machines and therefore being unable to work on other things. Machines down for maintenance can again lead to unsatisfied customers. All of these things add to the cost. In the most extreme cases, predictive maintenance can save lives[AP17].

Using predictive maintenance can help to improve the overall efficiency of a system. Predictive maintenance makes sure that maintenance is done when it is truly needed (or as close to it as possible)[LP17, HB11]. It is built on the concepts of "*if it is not broken, don't fix it.*"[Chu16]. That ends up utilizing the resources within a company in a more efficient and cost-effective way[LP17]. Advantages achieved by predictive maintenance are increased availability, quality along with increased productivity of equipment, while insurance and maintenance costs are decreased[eCCO12, AP17, IBM, SAP].

Hashemian and Bean[HB11] present an experiment which was done by engineers where they ran different systems at the same hard conditions, and where the remaining useful living time was different on the same components which were placed in different systems. In the worst cases, the system failed after 15 hours, while another system lasted up to 300 hours. This might lead to components being maintained after 15 hours of use. By utilizing predictive maintenance, the lifespan of these can be used more efficiently.

10.5 Other maintenance types

Horenbeek and Pintelon[VHP13] present a list of other regularly used maintenance types:

- *Block-based maintenance* - Independent of the failure history, a component is maintained at predefined times, which is repeated on a regular basis.
- *Age-based maintenance* - A component is maintained after a certain age or when it fails.
- *Inspection condition-based maintenance* - Inspection is done on a regular basis, and if a component seems to be below a set standard or value, it is maintained.
- *Continuous condition-based maintenance* - Sensors are used to monitor the state of components, and when the state of a component is below or close to a certain threshold, the component is maintained.

10.6 Summary

This chapter has given a literature review of predictive maintenance. During the literature review of predictive maintenance, the author's knowledge of the field has expanded massively and came up with the following definition

Predictive maintenance is monitoring of systems and components within them and utilizing history and real-time data to predict when one of the components or systems might fail.

By comparing it to other types of maintenance, it is clear why it should be used in an intelligent enterprise. Usage of predictive maintenance can help enterprises cut expenses and improve in different areas like increased availability and quality, safety, the productivity of equipment and facilities to name a few. Predictive maintenance can also help decrease maintenance costs and insurance costs. By having machines or equipment down for maintenance might result in unsatisfied customers and eventually loss of customers.

A downside of predictive maintenance is the initial cost and development of it. For many enterprises, it will be costly and time-consuming to develop a system which accurately can predict these events, and it requires a lot of data.
However, in the long run, it will be more efficient for enterprises to take advantage of this technology and tool.

11 Machine learning

In this chapter, a presentation of the basic concepts of machine learning will be given alongside different areas of use and a thorough presentation of the machine learning workflow. This was included in the preliminary study as the prototype would be developed using machine learning.

11.1 Definition

Machine learning takes advantage of pattern recognition and historical and real-time data to predict possible outcomes in the future.

11.2 Areas of use

Some problems in the world are very hard to fix with programming done by hand. In these use cases, machine learning is beneficial. Examples of such use cases are the detection of spam, face recognition, translation done by machines, speech recognition, and robot motion[Lis12]. Machine learning can also be used within medical diagnosis and biometrics[Kau16]. In medical diagnosis, symptom patterns can be turned into illness classification, while in biometrics recognition of iris, face signature and so on can be discovered. There are many more use cases than presented in this section and, as this thesis will show, it can also be used for predictive maintenance.

11.3 Machine learning workflow

There are many ways to describe the machine learning workflow[Ven17, Pug16, Wal15, Sci17], but at a higher level, they all are inspired by the same pattern as illustrated in figure 3. Swetha Kolalapudi has a video course about understanding machine learning and how to think about its algorithms[Kol16]. She presents a typical machine learning workflow which is also supported by other courses and presentations[Cha16, Kur16, Kur17]. The workflow consists of identifying which type of problem is at hand, accurately represent the data, apply a machine learning algorithm and then choose the model that gives the best result. This is a very generic view of the workflow, and it can be specified further as we will see in the following subsections.

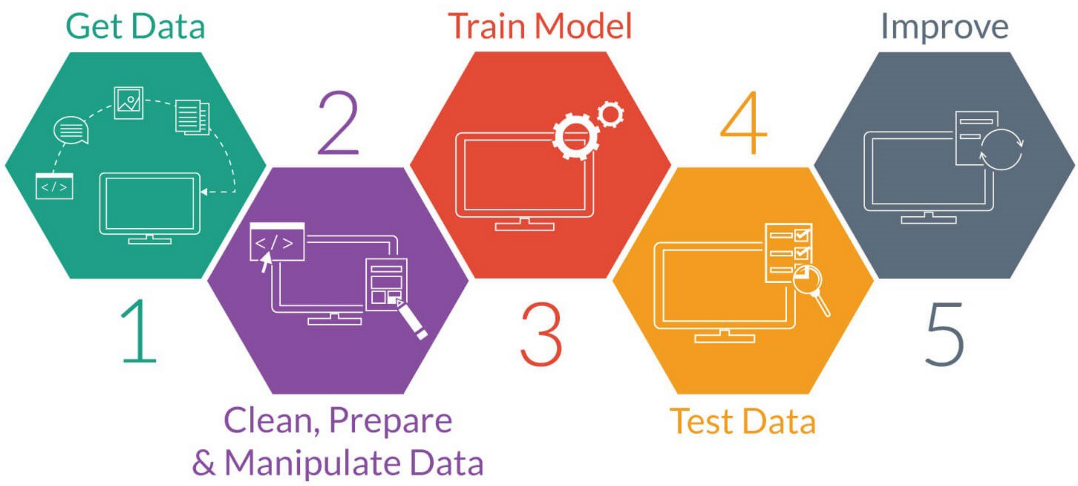


Figure 3: A typical machine learning workflow[Ven17]

11.3.1 Identifying the problem

The most important part of this step of the workflow is to understand what is the desired outcome. It is all about asking the right questions[Kol16, Kur16] to make sure the project is headed in the right direction. A problem definition can be set up by defining scope, target performance, a context for usage and how the solution will be put together[Kur16].

11.3.2 Data representation

While working with and trying to solve machine learning problems, enormous amounts of data are required to get the best possible result. The data can come from many heterogeneous sources and needs to be preprocessed before the different machine learning algorithms can handle the data. Several techniques can be used, such as data munging, feature extraction and dimensionality reduction[Kol16]. These techniques include fixing data that is either missing or corrupt and structuring of the data. Removing unimportant data columns are also relevant[Kur16] and called feature selection[Cha16]. A column is decided to be unimportant if it holds data which is not relevant to the machine learning training, such as an ID field.
An example of feature selection is if a data table contains a column for ID, which won't contribute to the results at all, and is for that reason removed.

Since the two first steps are the foundation of the next steps, it is natural that the most time will be spent within these two steps[Kol16]. A developer should also expect to return to these steps at a later point and restart the process based on trial and error processes[Kur16].

11.3.3 Selecting the algorithm

Several decisions need to be made when selecting a machine learning algorithm for a specific problem[Kur16]. First of all, there is a need to know the learning type to deal with for a project. Different types are supervised, unsupervised and reinforcement learning[Cha16]. Supervised learning is learning where the desired results are known beforehand, while for unsupervised learning this value is unknown prior to the training.

The outcome of the algorithm is also important, whether it is going to be a continuous value or a label. Furthermore, the complexity of the algorithm is a deciding factor. The different algorithms have different hyper-parameters which can be tuned to make the algorithm perform better given the specific dataset at hand[Kol16].

11.3.4 Training the model

As mentioned, machine learning needs a lot of data to generate results and especially good results. Training is done by running a lot of data through the algorithm which ends up producing a model which can predict new data based on the historical data used during training.

11.3.5 Testing the model

It is normal to set aside a part of the data at hand to use for testing at a later point. Testing makes sure that the model is not just specified for the exact data in the training set and are in danger of producing inaccurate results when the model is actually in use[Cha16].

Cross-validation is a technique where the initial dataset is split into a training set and a test set. Validation is done k times, and the model is trained and tested with the different splits[Kol16, Kur16]. Cross-validation is done to avoid overfitting and underfitting[Kur16].

11.3.6 Improving a model

If satisfaction is not acquired through the results, an algorithm offers several tuning parameters that can be done to improve a model[Kol16, Kur16]. Simply adjusting the current algorithm by selecting a new one might in some cases be the best option. If the decision is to stick with the same algorithm, then several things can be done including hyper-parameter tuning, getting more or improved data, fixing overfitting/underfitting by utilizing cross-validation and so on.

11.3.7 Using a model

After a model is completed, it can be utilized by calling it with the features that the model requires, and as a return value, the model will send back the predicted value based on previous training and tuning[Cha16].

11.4 Summary

This chapter has given a literature review on machine learning. The chapter has introduced a definition and presented some typical areas of use for the technology. The machine learning workflow was given a thorough walk-through, where steps presented were; identifying the problem, data representation, selecting the ML algorithm, training and testing of a model, and improvement and use of a model. At a later stage, this thesis will return to machine learning as the planned prototype will be developed using this technology.

Part V
Emerging and relevant technologies

In this part, a presentation is done of the results found by the literature review done on emerging and relevant technologies and approaches that exists for helping an enterprise enhance within each of the eight defined capabilities. This part is structured as such; each capability has its chapter where it is further discussed approaches and technologies for each of them. At the end of a chapter, a summary is given. A chapter is included to summarize the findings of this part.

12 Decentralization

This chapter will present approaches and technologies for *decentralization*.

12.1 Definition

"*A decentralized system is one which requires multiple parties to make their own independent decisions.*"[The]

12.2 Approaches

The idea behind decentralizing is to spread decision making[s.F02]. There are several advantages that comes with decentralization such as increased speed of decision making, increased productivity and quality of life, flexibility and motivation[s.F02, Mal04]. When employees understand the core values for an enterprise, there is no need for a top-down approach and decentralization can be used[Dou17].

One of the most critical tasks of a manager in decentralized enterprises is often to provide channels where knowledge can be shared between the different departments[Mal04]. An enterprise can gain an advantage by having the ability to communicate innovative ideas and creative information[Fre18] quickly. An enterprise does not want to end up with departments acting like data silos. Data silos are departments and similar which possess a lot of data, but the data is not being shared[YLL+09].

As reasoned in the *framework for proactive enterprises*, decentralization is an essential capability for a proactive enterprise. Before an enterprise decentralizes, some considerations need to be looked at. There are concerns about the value of the benefits of decentralization and if they will compensate for the cost of decentralizing. There is an initial cost to decentralize an enterprise, including setting up communication and channels for communicating and sharing knowledge between the different departments[WM96]. Other decisions that need to be made are setting the boundaries for the different departments. How much money out of the budget are they able to spend on creative innovations, will they be allowed to hire new full-time employees for their department without corresponding with the enterprise leaders. How much control do they have over the introduction of new products and which decisions can they take concerning business relevance[BSVR10].

Enterprises can decentralize on different levels as there exist several types of decentralization[s.F02]. *Devolution* is a form of decentralization where the higher parts of the hierarchy have almost no connection to the lower parts. Each level is independent and can make decisions to a certain degree. The higher levels set the boundaries they are limited by in the hierarchy. *Deconcentration* is another form of decentralization, and it is more strict than devolution. The idea behind deconcentration is to shift workload from central parts of the enterprise to smaller parts. The smaller parts are limited and are not able to make decisions, solely implement decisions made by the central parts. *Delegation* is a third form of decentralization. The idea of delegation is to shift decision making into smaller parts of the hierarchy. Within delegation, the smaller parts have authority to plan and implement decisions within boundaries. In either way, it is essential that all departments and parts work toward the common goals of the enterprise. Shared goals also help to bring people together and make results[Wun17].

Figure 4 shows three different structures of an enterprise where the single points represent a department within an enterprise. The leftmost illustration shows an enterprise split up into departments, where all of them work independently and they do not share information. The illustration in the middle shows a centralized enterprise, which has several departments but decisions are made from the top. The rightmost illustration shows an enterprise that is decentralized and has communication channels and shared information within the enterprise. George Wyner and Thomas Malone

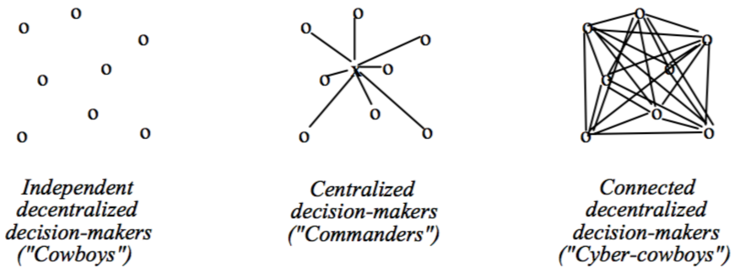


Figure 4: Different enterprise structures[WM96]

made the comparison of the different structures to cowboys and cyber-cowboys[WM96]. Cowboys on horses back act on their own and need to make their own decisions without knowledge from anyone else. Commanders make decisions for an entire troop based on information he receives from his scouts, while a cyber-cowboy makes his own decisions, which he can base on information and knowledge he receives from other cowboys.

Enterprises need to make a decision for whom they are going to decentralize to[s.F02]. The enterprise leaders need to decide which departments should have the ability to make decisions. As mentioned earlier, they also need to decide at which level the departments are allowed to make decisions and which tasks should be decentralized[s.F02].

To decentralize, it is important to understand and have enough knowledge about an enterprise and its goals[Mal04]. A manager needs to evaluate if the potential benefits of decentralization are essential to the enterprise if they can compensate for the potential cost of it and if the benefits will overcome the initial costs. The benefits of decentralization are positive, but in some cases, there might be better for enterprises to focus on other capabilities to improve.

There are some guidelines for when an enterprise should decentralize. When an enterprise is proliferating, or the industry itself is changing it can be smart to decentralize[Mal04]. While the enterprise is growing, introducing new departments and decentralize decision making will make it even more manageable in the future to continue decentralization. In a changing industry decentralization also makes it easier to keep up with the rapid changes and introduction of new departments or products. If an enterprise already has several small departments, that works independently, decentralizing decision making can help an enterprise by encouraging innovativeness. If an enterprise operates in a market with high competition, it is also an excellent decision to decentralize[BSVR10]. The same arguments as earlier are used to reason for this as well, as competition creates changing industries. Young enterprises which are close to the technological frontier are likely to decentralize, as they are in heterogeneous environments[AAL+07]. If there is a need for increased employee morale, decentralization can be the solution[Fre18]. Departments that are allowed their own decision making, and are in greater control of themselves can spark morale within the employees in the department. While being able to make their own decisions, they can often spur innovation and creativeness within themselves.

12.3 Technologies

To make decentralization work and for an enterprise to get the most out of it, communication and knowledge sharing is vital within the enterprise. In the following sections, there are presented some technologies and applications that are available for enterprises to make sure of communication and knowledge sharing.

12.3.1 Slack

Slack is a collaboration platform, and according to Slack's homepage[tea18b], their product unifies team communication and is making workflow better. Slack gives members the opportunity to search and find files, while also being able to search for colleagues and messages in the same place. In Slack, discussion and all communication are organized in so-called channels. It is normal to create several channels, one for each team, project and so on. All enterprise related documents can be shared within the channels and are accessible by other members of the same channel. By doing this, it is possible and easy to share knowledge with other employees. Slack offers voice and video call to its customers. It is also possible to share screens with other employees.

Slack has a rich set of options and tools. Slack integrates with a lot of other different services, such as Asana[4], Dropbox[5], Google drive[6] and others. Integration makes it a useful tool for more of its customers. However, for making an application like Slack successful in an enterprise, it is essential that there is an enterprise culture for it[Duf18]. Slack will not be useful if employees do not know how to use it, so an introduction course for employees might be a smart idea.

Slack is a way that the employees of an enterprise can communicate quickly and directly while also sharing knowledge which will be readily accessible to the other employees. Even when there are new employees, they will have access to the previously written messages and conversations that they are a part of. Employees can even customize notifications, meaning that they can specify a list of keywords, and whenever a message involves one or more of these words, they will get a notification[Duf18].

Slack is used by huge companies worldwide, such as AirBnb[7], Samsung[8], Google and Nasa[9] to mention a few. While Slack is one of the largest providers, they are also by far the most expensive. This is one of the reasons as to why enterprises might prefer their competitors.

12.3.2 Skype for business

According to Microsoft's homepage[Mic18b], Skype for Business is a powerful collaboration tool. They offer many services including recording of meetings that users have online, where each meeting can host as many as 250 users. An enterprise can, therefore, hold courses and meetings for their employees by utilizing Skype for Business. Microsoft values safety and privacy, so the video and audio streams are encrypted.

In earlier days, Skype for Business went by the names Microsoft Lync and Microsoft Office Communicator. Now Skype for Business is designed to enhance efficiency within an enterprise[Tay17]. The newest version lets employees work on the same documents at the same time, making knowledge sharing more comfortable and more accessible. There are many collaboration possibilities including Whiteboard document collaboration, Powerpoint document collaboration, and desktop sharing.

Skype for Business has also direct messages and group chats implemented, making it easier for employees to communicate and collaborate.

12.3.3 Microsoft teams

Microsoft Teams is another communication technology by Microsoft. It comes with a lot of the same features as Skype for Business, such as meeting recording, sharing of files and applications

[4]https://www.asana.com/
[5]https://www.dropbox.com
[6]https://drive.google.com
[7]https://www.airbnb.com/
[8]http://www.samsung.com/
[9]https://www.nasa.gov/

and implementation of private and public chat groups[Mic18a]. Microsoft Teams is included in the subscription plan that covers Office 365, which is widely used.

Microsoft Teams is designed explicitly for quick collaboration between employees through several types of media[Bra17]. It comes with strong security and privacy options which are essential for enterprises of all kind. The service also integrates with more than 150 third-party systems.

Microsoft Teams let their customers create groups with as many as 999 members. This will cover most enterprises and can, therefore, secure enterprise-wide communication channels and knowledge sharing which are vital attributes for a successful decentralization of an enterprise.

12.3.4 Alternatives

There are many alternatives to these team collaboration technologies[A.N17]. One of these alternatives is Bitrix24[10], which aims more for smaller companies and startups. They have a lot of the same features as Slack, and the others discussed, but the pricing is way lower than for example Slack's pricing plan. It comes with features such as tasks and project management, chat possibility through text and video and file management. Another alternative is HipChat which is also quite similar to the others as it includes private and group chat possibilities, file sharing and so on. Other alternatives are applications such as Hall, Pie and eXo Platform.

12.4 Summary

In this chapter, a presentation of decentralization, how an enterprise can decentralize and when an enterprise should do it has been done. Different considerations an enterprise needs to take before decentralizing have been presented, and also which benefits they can gain by decentralizing. As communication and shared information is key for decentralization, some technologies and tools that already exist have been shown, which enterprises can take to use to make decentralization more successful. Some of these technologies are more for knowledge sharing, and support decentralization indirectly by facilitating easier communication and information sharing.
It is important to understand that decentralization is not always the solution, even while being in heterogeneous environments. Centralized decisions might have to be made if resolving conflicts is important, when it is important to have much detail to a shallow level and when there is in fact not enough employees that can make good decisions[Mal04].

13 Interoperability

This chapter will present approaches and technologies for *interoperability*.

13.1 Definition

"*the ability of computer systems or software to exchange and make use of information.*"[Good]

13.2 Approaches

The capability interoperability, as defined in the *framework for proactive enterprises*, is about the internal interoperability within a company. For that reason, most of the interoperability functionality is "machine-to-machine" and concerns automated processes and linkage of applications[LW06]. Machine-to-machine interoperability mainly focuses on communication protocols and which kind of infrastructure is required to make those protocols work[RCL14]. It is about different programs being able to use the same exchange formats, use the same file inputs and outputs and that they can use the same protocols[Gis11].

[10] https://www.bitrix24.com/

There are several levels of interoperability[RCL14, CDV08]. Interoperability can come in several different degrees, starting at the lower level with data interoperability. At this level, it is about making different languages, and data models function together. Data interoperability involves being able to find and share data that is gathered from heterogeneous sources. Data interoperability covers techniques such as semantic data representation, data standardization, schema matching and data mediation. A lack of standardization during the design phase and development phase of a program can very well lead to the lack of interoperability[Gis11]. The next level embraces services and is called service interoperability. At this level, the aim is to make applications and services that were developed independently to be able to work together and exchange information and services between each other. The third level is called process interoperability, and its goal is to increase cooperation between processes. The processes should be able to use each other and thereby deliver services to each other. To clarify the difference between a process and a service; a process is several services or applications connected. A fourth level is called the business level and is the interoperability between businesses.

Several approaches are defined for interoperability[RCL14, CDV08]. These approaches are defined for both enterprise internal-interoperability, but also for cross-enterprise interoperability. The three different approaches are integrated approach, unified approach, and federated approach. These three differ in a manner of how strict the rules of the cooperation using interoperability are. For the integrated approach, there exists a common format which models are to be made out of. This needs to be followed strictly, and before work, the format must be agreed upon by all involved parties. Some examples of standards that can be used are HTTP, HTML or XML[CD03]. The second approach is the unified approach where there exists a common format, but not a strict one. At this approach, there is a need for mapping between models for interoperability. The third and final approach is called the federated approach, and in this approach, no common format is defined. This is the weakest form of the three approaches, and modifications need to be made continuously, but it also means that there is no need for agreeing on a common format. Making adjustments converting the interface of one product into another product's interface can be solved by using a "broker" or a middle-ware[CD03].

There are different ways to achieve interoperability within software[Gis11, RCL14]. Product testing is a technique, where different products that are developed using common standards are tested to see whether the interoperability was successful. Another technique is product engineering, where products are developed using the same standards and that are designed to interoperate with other products.

When an enterprise grows and consists of several departments, services, processes and systems it might be time to enhance interoperability. Having systems that work together efficiently can spark collaboration processes and open innovations[WMC$^+$16]. Interoperability can be planned at any time in the life of an enterprise, but the earlier it is implemented, the easier it is to maintain. When an enterprise is in an ever-changing environment, interoperability will make it possible for the enterprise to solve organizational challenges faster[MS16, WMC$^+$16], and thereby compete.

Several challenges come with interoperability. Some of them are the ability to process data from heterogeneous sources and information systems; another is to have the resources required to integrate other systems[LW06]. Because of the difference in formats and standards between different systems and technologies, there is a challenge for handling data sent between systems. Often there is no possibility for "instant integration." A technological barrier can be such as incompatibility between information technologies, such as infrastructure, platforms and architecture[RCL14, CDV08]. Due to an oversupply of standards and heterogeneous platforms for hardware and software as well as other reasons, interoperability has become a challenge for enterprises[CD03].

13.3 Technologies

There are developed several interoperability frameworks which contain concepts, assumptions, and practices which can help with interoperability issues. There is a definition of interoperability frameworks which is made by The European Interoperability Framework, and it is as follows:

> *"An interoperability framework can be defined as a set of standards and guidelines that describes the way in which organizations have agreed or should agree, to interact with each other. An interoperability framework is, therefore, not a static document and may have to be adapted over time as technologies, standards and administrative requirements change"*[RCL14]

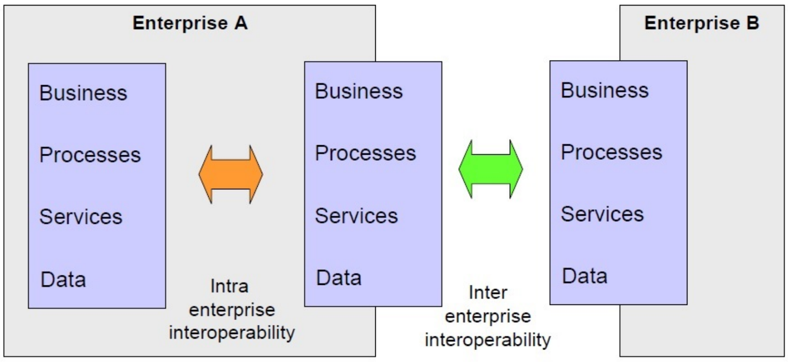


Figure 5: Interoperability within an enterprise and with other enterprises[Wik18]

Figure 5 shows the different levels of interoperability both within an enterprise and towards other enterprises. In the coming sections, we will look at different interoperability frameworks.

13.3.1 IDEAS

The Interoperability Development for Enterprise Application and Software (IDEAS) Framework is a framework that was developed by the IDEAS project[CDV08, CD03]. The framework aims to reflect upon how to achieve interoperability on different levels, and IDEAS defines interoperability as the capability of interactions between the software applications at an enterprise[DCV05]. The framework has several different layers which are the application layer, the data layer, the business layer, the knowledge layer, and the communication layer. These layers help enterprises make decisions, operate and exchange information between each other[LW06]. It is stated that interoperability is achieved if interoperability is fulfilled in at least three levels[DCV05]. In figure 6, we can see the IDEAS framework presented where each of the layers are present.

	Framework 1st Level	Framework 2nd Level	ONTOLOGY	QUALITY ATTRIBUTES			QUALITY ATTRIBUTES		
			Semantics	Security	Scalability	Evolution	Performance	Availability	Portability
ENTERPR. MODEL	Business	Decisional Model							
		Business Model							
		Business Processes							
	Knowledge	Organisation Roles							
		Skills Competencies							
		Knowledge Assets							
ARCHITECT PLATFORM	Application	Solution Management							
		Workplace Interaction							
		Application Logic							
		Process Logic							
	Data	Product Data							
		Process Data							
		Knowledge Data							
		Commerce Data							
	Communicat ion								

Figure 6: IDEAS Framework[RCL14]

13.3.2 ATHENA Interoperability Framework

The ATHENA Interoperability framework is a framework that was developed by the ATHENA integrated project[CDV08, BEF$^+$07]. The framework was developed for software systems and enterprise applications[LW06]. The ATHENA framework has a different aim than the IDEAS framework in a way where its aim is to analyze and understand requirements and business needs. The framework also aims to provide a model-driven solution approach to interoperability problems. It reaches its aim by providing guidelines, principles, and patterns, which can all be used for interoperability problems.

Similarly to the IDEAS framework, the ATHENA framework also has application, data and communication environments. The framework offers tools and communication platforms for the development of systems and enterprise applications[LW06].

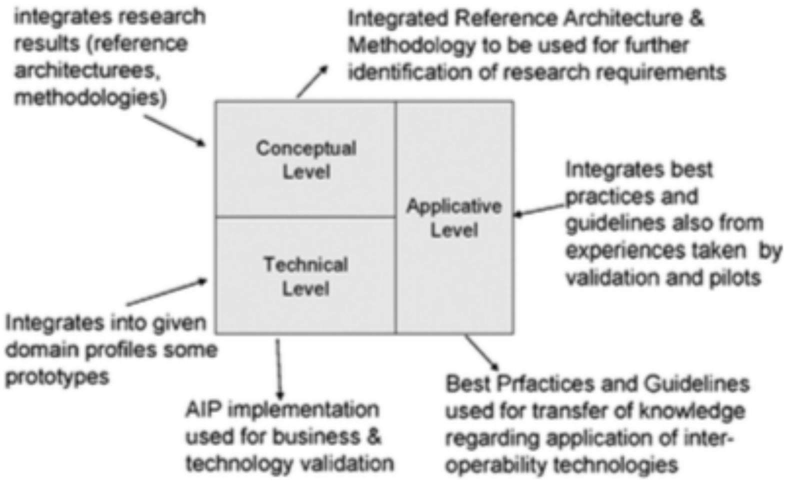


Figure 7: ATHENA Interoperability Framework[RCL14]

Figure 7 presents the ATHENA framework and the different levels of it, which are the conceptual, the applicative and the technical level. The focuses within the three different layers differ, where the conceptual level is focused on concepts, models, and metamodels, the applicative level focuses on methodologies, standards, and domain models, and lastly, the technical level focuses on technical development and the ICT environment[Pet14, JKL07].

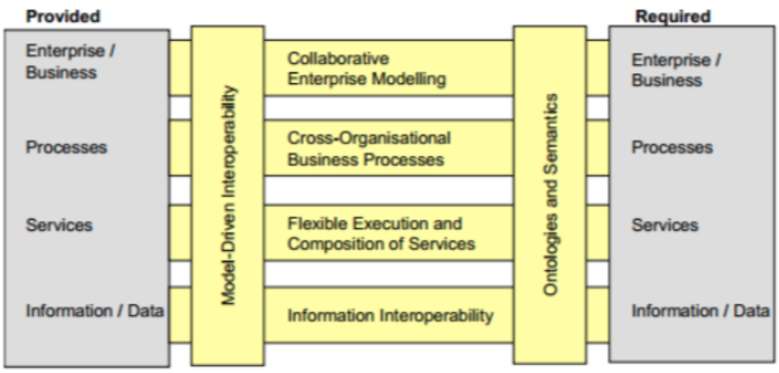


Figure 8: Conceptual level of ATHENA Framework[Pet14]

Figure 8 presents the conceptual level of the ATHENA framework. The same levels as discussed in section 13.2 can be seen on each of the gray boxes. The conceptual level provides a bottom line for putting different aspects of interoperability into system[Pet14].

13.4 Summary

This chapter has presented and furthermore explained the capability interoperability from the *framework for proactive enterprises*. Approaches and different levels of interoperability have been presented, and the four different levels of interoperability are data, services, processes, and business. To collaborate with other enterprises using technology, interoperability is a prerequisite. It is stated that when an enterprise is growing, interoperability becomes more important, and focusing on interoperability should be done from the beginning of. Following standards or using brokers can help maintain and achieve interoperability. Under section 13.3, two interoperability frameworks were presented, which can guide enterprises in the hunt for interoperability. Other frameworks also exist, but was not discussed in this thesis such as the Enterprise interoperability framework[Che06] and the LISI reference model[$G^{+}98$].

14 Awareness

This chapter will present approaches and technologies for *awareness*.

14.1 Definition

"*knowledge or perception of a situation or fact.*"[Goob]

14.2 Approaches

As described in the *framework for proactive enterprises*, there are two types of awareness an enterprise needs to consider. Internal awareness includes the factors within enterprise borders, such as technical infrastructure, learning capabilities, organizational structure but also system awareness. The other type of awareness important to an enterprise is external or environmental awareness which includes market or collaboration opportunities and environmental threats amongst others. Being aware increases the readiness for change for an enterprise[BDLB14].

A SWOT analysis is a tool used by businesses to identify strengths, weakness, opportunities and threats[Win18, HW97]. A SWOT analysis helps increase both the internal and external awareness by looking at the internal structure and the external environment. A SWOT analysis can also be referred to as an environmental scan because of its nature[SCK03]. Table 19 shows a setup for a SWOT analysis.

Internal assessment	**Strengths** Where we can outperform others	**Weaknesses** Where can others outperform us
External assessment	**Opportunities** How we might explore the market	**Threats** What/who might take our market

Table 19: A setup for a SWOT analysis[SCK03]

SCOPE is a planning method where several factors are taken into account. These factors are the situation, core competencies, obstacles, prospects, and expectations[Win18]. Utilizing SCOPE planning gives an enterprise a wider awareness than only using a SWOT analysis as it can provide more information. As mentioned there are five different factors[Riv16]. Situation goes for the background scenario an enterprise potentially can have. The core competencies mean the strengths that an enterprise possess internally that can give them a better position externally, while obstacles are the external factors that may threaten an enterprise and its position. Prospects and expectations are both about outcomes where prospects are favorable outcomes and expectations are realistic outcomes.

SOAR is an analysis tool that focuses on the positive. It does not look at the weaknesses and threats at an enterprise. It does so by identifying strengths, opportunities, aspirations, and result[Win18]. Table 20 shows a setup for a SOAR analysis.

Strategic inquiry	**Strengths** What are our greatest assets	**Opportunities** What are the best possible market opportunities
Appreciative intent	**Aspirations** What is our preferred future	**Results** What are the measurable results

Table 20: A setup for a SOAR analysis[SCK03]

There is another approach for staying environmentally aware[BDLB14]. It bases itself on three general steps which include that threats are identified, understanding the threats and the sources of the threats and lastly the size of a threat they impose on an enterprise is defined.

An enterprise should be continuously aware, both internally and externally to detect changes fast. Being continuously aware of environmental changes puts an enterprise in a good position concerning acting on them. An enterprise needs to be aware of changes and new opportunities in the environment and market that it operates in. This is not always easy. The approaches described are static methods where actions are needed to be taken.

14.3 Technologies

The last section presented approaches for analyzing an enterprise to increase awareness both within the enterprise borders and outside of them. This section will look at some techniques an enterprise can use to take advantage of technology to be aware of the situation they are in.

14.3.1 Collection of sensor data

An enterprise can use technology to stay more aware of their situation, and especially for internal awareness, sensors can be used to capture an enterprise's situation[Has11]. Gathering information in itself is not enough to increase the awareness, so such an approach must be combined with other capabilities such as perceptivity, which is explained in chapter 15. For example for enterprise machine statuses, sensors and IoT technology can be used to gather information about components which can be used by intelligent systems to understand the situation for each machine[HB11].

In the *framework for proactive enterprises*, there is included internal system awareness where it is specified that this means that sensors are used to capture the real-time state of components in enterprise systems. The use of sensors to capture this kind of data is the base foundation for predictive maintenance as it gives enterprises real-time opportunities to view the current state of all the components in their systems and based on the data collected, decisions can be made to act upon a bad state[KG09]. As we will see later in part VI, Business A is using sensors in their enterprise machines to monitor the different components in their machines continuously. The machines send status reports to the enterprise servers which can then handle the data.

Figure 9 shows an example of where sensors are used to monitor components in a machine. One sensor each is placed on three different vital components of the machine. Each of these sensors monitors the state of the component they are assigned based on which type of sensor it is. The sensor data is collected and sent to a cloud service which can present it to the enterprise.

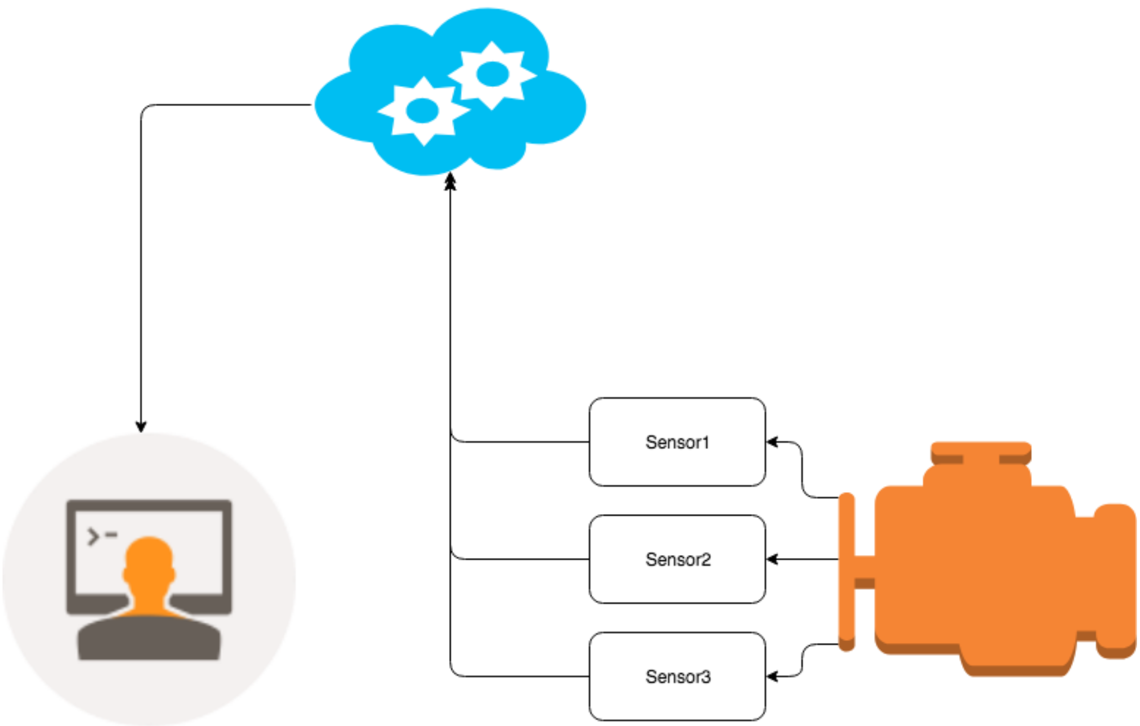


Figure 9: Sensor technology

14.4 Summary

This chapter has presented approaches and technologies connected to awareness of an enterprise. There has been a presentation of several tools and approaches that could be used by an enterprise to increase both the internal and external awareness such as SWOT, SOAR, and SCOPE among other. The chapter stated that an enterprise should be continuously aware of their internal and external situation, and concerning technology, a way to increase awareness is to use and take advantage of sensor data. For increasing internal system awareness, collection of sensor data was presented as a technology that can be used to accomplish this.

15 Perceptivity

This chapter will present approaches and technologies for *perceptivity*.

15.1 Definition

"*having or showing sensitive insight.*"[Gooe]

15.2 Approaches

As enterprises gather and generate enormous amounts of data, it is vital that they be able to make meaning out of it by analyzing it and making it available for the intelligent systems or managers who make decisions. This will make it valuable for them, and they can explore and find new opportunities through analyzed data, but this requires that the enterprises be able to handle and store big data[MPCAF15]. There are two different approaches to data processing, which is historical and real-time analysis[MSB+14b, WMC+16, LAK+15]. These approaches can also be called in-batch processing and in-stream processing[ZWLS15].

To fully take advantage of data, it is essential to be able to handle, organize and analyze data from heterogeneous sources. Part of the data gathered will be raw business knowledge, which will need to be transformed into business intelligence[LAK+15]. Extracting value out of data is time-consuming and can be difficult. There are several aspects of the value of data that can be extracted, such as actionable knowledge, return on investment, business value and how it can be of value for a customer[SS17].

For doing real-time data processing, several layers can be used[ZWLS15]. These are the data and analytics layer. The data layer consists of cleansing and gathering data, while also including the storage of data. While an enterprise is cleaning the data that it processes, it is important that they do not filter away too much useful information by making the granularity too small. It is also essential that they do not let too much irrelevant information pass the cleaning process[ZWLS15]. The analytics layer's primary purpose is to do some fundamental data analysis, preparing data for the intelligent systems. The analytics layer can also prepare the data for managers to aid them in decision making. There are several ways that the analytics can be done, some techniques are SQL queries, data mining, statistics, decision analysis, optimization, visualization and machine learning[SS17]. When data processing is done, the results can be handled in several different ways. If an enterprise possesses intelligent systems, then the data can be prepared in such a way that it is ready to be sent to the intelligent systems. The intelligent systems can then help managers make decisions or make decisions themselves based on the data they are given. In case of a lack of intelligent systems, the processed data can be visualized for managers and thereby help them make decisions off of the data. The data can be presented to managers in the form of graphs, reports or similar[SS17].

An enterprise should invest in data analysis tools when they generate or need to process vast amounts of data where they want to find hidden patterns or assist their managers in some way when it comes to decision making.

15.3 Technologies

There are a collection of tools that can be utilized when it comes to perceptivity and making meaning of data gathered. The following subsections will present some of them.

15.3.1 Apache Spark

Apache Spark is an analytics framework that is designed to handle large-scale analytical processing[SS17, ZWLS15] fast. It has been an Apache top project since February 2014 and is an open-source project that over 30 companies have contributed to through more or less 150 software engineers. Apache Spark is used for fast and repeatedly (near) real-time processing of data[SS17]. Spark works well for machine learning and is designed to be able to work on top of the Hadoop Distributed File System[SKRC10]. Since the framework uses in-memory technology, it can query between one and two terabytes of data in one second, promising speeds up to 100 times faster than Hadoop Map/Reduce. Apache Spark supports ETL (extract, transform, load), machine learning, graph computation and stream processing[SS17].

Examples of how to use Apache Spark can be found here:

https://spark.apache.org/examples.html

15.3.2 Hadoop Map/Reduce

While Spark is more designed for (near) real-time analytics, Hadoop Map/Reduce performs well for batch applications and was created to analyze big data[SS17]. Hadoop is designed in a way where it can combine and handle unstructured data in several ways to prepare it for data mining or other analytics tools. Tasks are run in parallel on thousands of nodes[Had13] and the framework handles scheduling of tasks, as well as overlooking tasks, and if some tasks fail, the framework makes sure to try to re-run them.

When it comes to big data analytics, Hadoop has become one of the big standards, but in its earlier versions, Hadoop struggled with performance problems[DQR12]. These problems are now history and Hadoop can be used with many techniques to boost performance. As Apache Spark, Hadoop is also open-source and is a version of the algorithm Map/Reduce[CZW+16]. By utilizing the Map/Reduce algorithm, Hadoop can do automatic parallel data processing.
Figure 10 shows how the MapReduce algorithm works and presents how Hadoop MapReduce can be used to analyze data. The output of the algorithm is preprocessed data.

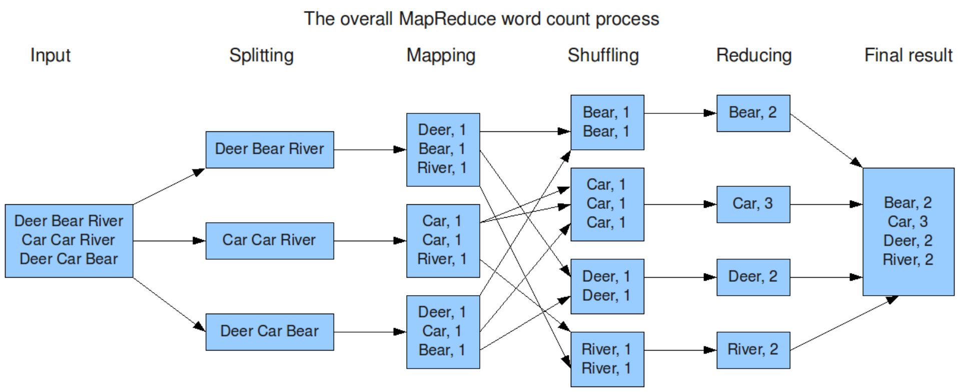


Figure 10: MapReduce algorithm[Lă17]

A tutorial for using Hadoop Map/reduce can be found here:

https://hadoop.apache.org/docs/r1.2.1/mapred_tutorial.html

15.3.3 Amazon Elastic MapReduce

Amazon EMR is a product delivered by Amazon Web Services. The service makes it easier to process vast amounts of data in a quicker and more cost-effective manner[VM14]. Some of the use cases that Amazon EMR can handle are data transformation, machine learning, web indexing and financial analysis amongst others[Ser18a]. Amazon EMR is easy to use, and a cluster can be up and running within minutes. Costs generated by Amazon EMR only comes through what an enterprise use and using the Amazon Web Services security settings, the cluster is secured.

EMR can be configured to use with Hadoop to distribute the data analytics across a cluster of Amazon EC2 instances, while another option lets an enterprise use EMR with Apache Spark[Ser18a]. Several other tools can be used in combination with Amazon EMR which makes it user-friendly and flexible.

How to get started using Amazon Elastic MapReduce can be found here:

https://aws.amazon.com/emr/getting-started/

15.4 Summary

This chapter presented approaches and technologies for the capability perceptivity from the *framework for proactive enterprises*. The chapter stated that for enterprises to take advantage of the data they have, it is crucial that analysis is done to fully understand what the data can be used for and what more information it contains. It was also stated that an enterprise should focus on perceptivity when they possess a vast amount of data and would like to take advantage of the information it holds. Technologies such as Apache Spark, Hadoop Map/Reduce and Amazon Elastic MapReduce was introduced shortly to guide enterprises in the direction of technologies that can be used for data analytics.

16 Prediction

This chapter will present approaches and technologies for *prediction*.

16.1 Definition

"*say or estimate that (a specified thing) will happen in the future or will be a consequence of something.*"[Goof]

16.2 Approaches

Being able to predict outcomes is an increasing capability within computer science[Dha13]. Several approaches can be used for predictions, and one of them is machine learning. Since there was done a more in-depth literature review of machine learning before starting the master thesis, it is referred to chapter 11. The reason this literature review was done prior to the work of the master thesis itself was that of the development of the prototype presented in part VI using machine learning.

Complex event processing (CEP) is another approach that can be taken for prediction. Vera Goebel stated a definition for complex event processing;

"*event processing that combines data from multiple sources to infer events or patterns for complicated situations.*"[Goe16].

The definition shows that CEP uses information from several sources to combine them and predict outcomes. CEP is already integrated and makes a difference in different areas such as energy management, logistics, manufacturing, and finance[BK09]. CEP systems are robust and handle a high throughput of events in near real-time[Tec12].

The input to the CEP systems are events, and an event is described as such by David Luckham; "*an object that is a record of an activity in a system*"[Luc02]. An event can be described as a change of state of an object. The events that come in are lower-level events which means that they have information of certain things at specific points in time. The output of CEP systems comes from pattern matching[Rob10] and are complex events. Complex events consist of information from several lower-level events and are also called high-level events. During the pattern matching, events are filtered away if they are not relevant for the output desired, but higher-level is also created by using information from lower-level events.

Combining CEP with automated decision makers would be a way to improve prediction and intelligence within an enterprise[Rob10]. Decision makers, whether they are automated or people, could react upon the outcomes of the CEP, and thereby make decisions faster, concerning changes that have been made. This improves the proactive capability of an enterprise.

Figure 11 shows an example of how CEP technology can be used for predictive maintenance. The CEP system receives events from sensors registered with an enterprise machine and based on the input; the CEP system predicts a new event which is presented to the decision makers so they can react to it. An example of how CEP can be combined with predictive maintenance can be found in a article by Petersen et. al[PvdKP16] where they introduce predictive maintenance for the production line of an enterprise.

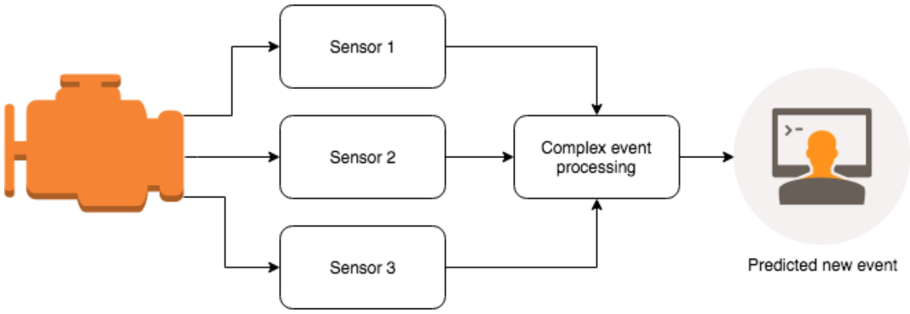


Figure 11: CEP and predictive maintenance

Figure 12 shows a process for complex event processing, where there are five blocks. The first block represents the event input, which can consist of several different sources including, amongst others, databases, feeds, event stores, and web services[Tec12]. The preprocessing block cleans and prepares the data for the next step which is the pattern matching. The preprocessing step is essential as every type of event input might have its format, but to make a general pattern matching there is a need for the same format on all events. Post-processing is done to make the output data fit back into the event databases or targets, meaning that the events go back to there original format. Only the events that make it through the pattern matching step makes it to the post-processing step.

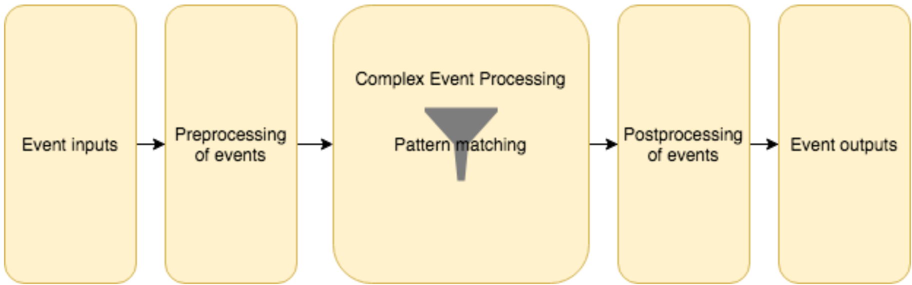


Figure 12: Complex Event Processing

When an enterprise is in need of being able to predict the future or they want to be able to save money through predictions they can start using prediction technologies to achieve that. As stated in chapter 10, for example, predictive maintenance can save an enterprise's money in the long run. With today's technologies, starting to utilize prediction ability is not necessarily so expensive and does not require expertise in the field.

16.3 Technologies

The following sections will present some technologies that can be used for prediction. Amazon Sagemaker and Apache Spark both use machine learning to make their predictions, while Oracle Complex Event Processing and StreamBase use complex event processing and pattern matching to do the same.

16.3.1 Amazon Sagemaker

Amazon Sagemaker is a machine learning tool delivered by Amazon Web Services[AWS18]. It is available as a service on the AWS platform. Sagemaker is designed to make it simpler for developers and computers scientists to train and deploy machine learning models for their use cases[Ser18c]. While using Sagemaker, there are typically three stages that need to be handled; generate example data, train a model and deploy a model. While doing machine learning, a huge part of the work is done by preparing data. This is typically done in three stages[AWS18]; fetch, clean and prepare.

First, the data needs to be fetched from somewhere. It might be a public dataset or a private one. The next step, cleaning the data, will consist of making the data consistent if different values are used for the same attribute or if some values are missing and so on. This is all done to have a better prediction model. Data preparation can also be done where for example attributes are combined to make a better prediction model.

For training the model, a machine learning algorithm needs to be used. Amazon provides a set of out-of-the-box solutions that can help the developers. Evaluation of the model that is created through the training can be done through Sagemaker as well.

Deploying the model and taking it into use is comfortable with Sagemaker as they let the developers host an endpoint where data will be sent in, and a prediction will come back.

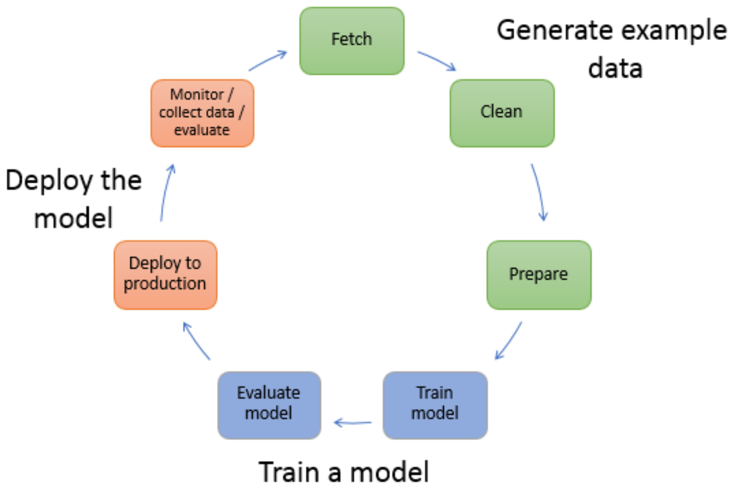


Figure 13: Machine learning workflow on Amazon Sagemaker[AWS18]

16.3.2 Apache Spark

A brief introduction for Apache Spark is done in section 15.3.1, but then for the capability perceptivity. An explanation of how this technology can be used in concern with prediction will be given. In addition to data analysis and data processing that Apache Spark can handle is also iterative machine learning tasks[MBY$^+$16]. Spark has an open-source library for machine learning which is called MLlib[11], which includes a range of machine learning algorithms and features to be used for machine learning. The library has more than 50 common machine learning algorithms that are included[ZXW$^+$16].
One of the clear advantages of Apache Spark and machine learning is the integration with data analysis and processing. This combines the data preprocessing that is required before putting data into a machine learning algorithm and the actual training.

16.3.3 Oracle Complex Event Processing

Oracle Complex Event Processing is a low-latency and complete solution for building event-driven applications[Cha06]. Oracle CEP is based on Java and build applications which can filter and process events in real-time[Gui, Tec12]. By utilizing technology like Oracle CEP, enterprises can have their applications run real-time intelligence and take advantage of rapid predictions based on patterns.

[11] https://spark.apache.org/mllib/

Based on user-defined rules, Oracle CEP can connect a vast amount of data to events which can help decision making for an enterprise[Gui]. Typically, the amount of events that are constructed during the complex event processing is far less than the number of input events that are taken into the system. The reason is that many events are filtered away as not relevant.

16.3.4 StreamBase

StreamBase is a product delivered by TIBCO Software Inc.[12]. The tool is high-performance and can be used to build applications for analyzing and reacting to real-time data[Inc18, Nam15, Tec12]. It is based on complex event processing and can help enterprises with predictions. By utilizing StreamBase, enterprises can set-up real-time systems quickly and reduce the cost and time of deployment that the components can promise[Tec12].

The tool supports an enterprise in the detection of significant events and can help them react to them in real-time. By reacting to it, an enterprise can adjust to the changes and update processes based on information provided by StreamBase and by business goals. IoT brings much potential when it comes to gathering and analyzing data, and StreamBase can help an enterprise process all of this information.

16.4 Summary

This chapter has presented the capability of prediction defined in the *framework for proactive enterprises*. The chapter presented different approaches when it comes to prediction which were machine learning and complex event processing. Machine learning had been presented earlier in the thesis, and this chapter, therefore, refers to it, while complex event processing gets its introduction here. Two technologies were introduced for machine learning which were Amazon Sagemaker and Apache Spark with the MLlib library. For complex event processing, there were also two technologies that were presented, and they were Oracle Complex Event Processing and StreamBase. These technologies can be experimented with by enterprises which want to start using prediction within the enterprise.

17 Intelligence

This chapter will present approaches and technologies for *intelligence*.

17.1 Definition

"*the ability to acquire and apply knowledge and skills.*"[Gooc]

17.2 Approaches

Decision support systems (DSS) are systems that can aid managers or other responsible decision makers with decision support[YTLT01]. The system itself can suggest a solution to a given problem based on the information it has, or it can help display information in a new way so that decision can be made easier by the ones in charge. There are mainly five different categories of DSS; data-driven, model-driven, knowledge-driven, document-driven and communication-driven[Pow00, PS07].

An approach that can be used for DSS is web-based DSS. A web-based DSS is a system that can help with decision support based through a web-browser[Pow00], which makes it accessible from more locations and for more people, in some cases including enterprise customers[Pow00]. A web-based DSS can implement any of the previously mentioned categories of DSS[PS07]. Such an approach for this kind of system can retrieve and analyze data by accessing large multidimensional or relational databases that the enterprise has. The system can also access trained prediction models which can

[12] https://www.tibco.com

contribute to the decision support. The fact that the decision support is available online through a browser has helped reduce previous barriers such as accessibility.
Figure 14 shows a typical architecture for a web-based DSS. The system is accessed through a browser, which makes it accessible for more people. Through the server, the browser has access to the model(s) which can provide decision support.

Figure 14: Architecture for a web-based decision support system

A type of DSS is as mentioned model-driven DSS. It does not require vast amounts of data to work properly[Pow00] in contrast with prediction models made through machine learning. The model-driven DSS can present its solutions based on information that is given to it by the responsible decision maker, and the information can easily be altered so that a decision maker can see the sensitivity of an outcome[PS07]. A model-driven DSS provides its decision support through uses of models such as algebraic, simulation, financial and optimization to mention some[PS07]. Another type of DSS is data-driven DSS[Pow08]. A data-driven DSS has another approach where it takes more advantage of real-time and historical data that an enterprise has either internally or externally[Pow00]. An example of data-driven DSS is business intelligence systems[PS07]. Easy access to data is key to having a successful data-driven DSS[Pow08].

Intelligent decision support systems are systems that take advantage of artificial intelligence to enhance their decision support[FVC15, GBM00]. Intelligent DSS is natural to combine with data-driven DSS as they contain a lot of data which can be utilized by artificial intelligence. The intelligent DSS should also be able to provide feedback to improve the prediction model. Prediction models can be obtained by using the technologies presented in chapter 16. In an article by Fernandes et al[FVC15], they provide an example of an intelligent DSS that can help predict the popularity of news articles before they are released. For an intelligent DSS to work correctly, it is essential that domain experts help prepare the data before it is given to the system[YTLT01].

17.3 Technologies

The technologies that can be used for intelligent decision making are similar to the ones that can be used for prediction. Machine learning and artificial intelligence are techniques that are used for supporting decision makers in making a decision. Complex event processing is also a technology that can be used and for the technologies that are relevant to the intelligent capability within the *framework for proactive enterprises* it is referred to chapter 16.

17.4 Summary

In this chapter, several approaches have been presented to the intelligent capability defined in the *framework for proactive enterprises*. The intelligent systems in an enterprise need to be able to make smart decisions based on the input they get from either the colossal amount of data they can process or through prediction systems if an enterprise has those. Decision support system (DSS) can aid decision managers or the systems themselves can make the decisions. There are in general five different types of DSS, which are data-driven, model-driven, knowledge-driven, document-driven and communication-driven[Pow00, PS07]. All of these types of DSS can be implemented through web-based DSS which makes the systems more accessible to decision-makers and stakeholders. Intelligent DSS are decision support systems that utilize artificial intelligence to make smart decisions. Some of the technologies that can be used for intelligent decision support systems are through machine learning and complex event processing and are presented in chapter 16.

18 Action

This chapter will present approaches for the capability *action*.

18.1 Definition

"*the fact or process of doing something, typically to achieve an aim.*"[Gooa]

18.2 Approaches

There are several ways an enterprise can act upon the information that has been processed, analyzed and handled. An enterprise can possess automated processes that can perform actions based on rules decided[FAB+11]. This means the process can be done without human interaction. Another way an enterprise can act is with human interaction. Based on the intelligent systems described in the earlier sections, reports and visualizations can be presented for managers and reports can be generated and sent to the decision making responsible. The intelligent systems can propose different solutions, or they can decide a final solution that should be implemented or acted upon. The actions that are taken, either manually or automated, needs to be aligned with the business goals the enterprise has[PvdKP16]. The action phase should also contribute with feedback to the intelligent systems on the results and consequences of the actions[CRVP01]. This will secure continuous learning and keep the intelligent systems improving.

In addition to how actions are performed, either through human interaction or automated processes, there is another aspect of acting. The time that actions are done is crucial for enterprises and can mean a big difference for them. Actions can be taken reactively, meaning nothing is done until it is in fact needed. An example of this could be if maintenance is done on a machine when a component fails, or other failures occur which takes the machine out of service. The other way an enterprise can act is proactive, which means actions are taken before something happen. An example would be to perform maintenance on a machine that is still working, but maintenance is done due to predictions that a machine will go out of service in the near future. Unplanned stops can then be prevented. To clarify, proactive actions are done ahead as an enterprise is able to predict the problem and take timely action to minimize risks and damage. Technologies such as machine learning and complex event processing (explained in chapter 16) can help an enterprise become proactive.

An example of how automated actions can be helping an enterprise is the introduction of an automated production line. Several advantages can be achieved when automating a product line within an enterprise such as low unit cost, automated material handling, and high production rate[Zha14]. Other advantages are that human resources are freed up so they can focus on other tasks and improve. Some disadvantages are that an automated production line typically requires a high initial investment and that the product line can be less flexible for product changes[Zha14].

18.3 Summary

This chapter has looked at approaches for the capability action defined in the *framework for proactive enterprises*. The chapter states that actions can be taken either automatically with machines and robots or through human actions. The actions taken should provide feedback, so it is possible to learn from them and use this information to improve as an enterprise. The chapter also discussed the time of when actions can be taken in an enterprise, where the main difference was in whether actions were taken reactively or proactively.

19 Extroversion

This chapter will present approaches and technologies for *extroversion*.

19.1 Definition

"*The ability to open boundaries and collaborate with other enterprises*"

19.2 Approaches

Extroversion as defined in the *framework for proactive enterprises* was about how an enterprise opened its boundaries and welcomed collaboration with other enterprises. It was about how they could go together to make collaboration networks to be able to deliver products they usually would not be able to do by themselves. With extroversion, enterprises will be able to share knowledge, resources and sometimes they might even share risks[PvdKP16]. To be able to be extrovert, there is a necessity of a willingness to interoperate or communicate and collaborate. Approaches that can be taken to ensure extroversion are open innovation and co-creating with customers. Extroversion is not necessarily across organizations, but it can also be with customers or the whole value chain.

Being able to achieve extroversion can generate numerous benefits[RCL14], such as successful collaboration processes and thereby probable new markets to explore and take advantage of. Other benefits are improved efficiency as well as more accessible information[WMC$^+$16].

Extroversion faces some challenges[LW06]. There is a need to be able to create a win-win-situation, meaning that all enterprises involved will gain on the collaboration. Enterprise partners not trusting each other is also a challenge. Questions such as who is responsible for what and who has the authorization to do what can be asked to clear things[CDV08].

Social technologies are an approach that can be taken to be extrovert. Social technologies are defined as "*software that supports the interaction of human beings and production of artifacts by combining the input from independent contributors without predetermining the way to do this*"[SN08]. From this definition it is possible to see that collaborators in social technologies not necessarily know each other[MWBR11]. This can lead to increased collaboration and knowledge sharing, as well as peer-to-peer learning[ACZ13]. Social technologies have had an exponential growth in recent years with potentially hundred of millions of contributors[MWBR11], and there is also an increasing interest in sharing professional achievements through social technologies[Yan08]. Other advantages are improved knowledge exchange, application of situational context, increased process transparency and process requirements integration, and examples can be wikis and blogs amongst others[MWBR11].

19.3 Technologies

This section will present some technologies that can be used to work towards extroversion. The technologies presented are collaboration tools that can be set up between several businesses.

19.3.1 Microsoft Sharepoint

Sharepoint is a team collaboration tool provided by Microsoft. It has support for document management, which gives opportunities such as reviewing, editing and knowledge sharing[RL02, DR13]. The tool enable employee collaboration not just within enterprise borders, but also cross-enterprise. Enterprises are able to collaborate cross-enterprise through the creations of so-called *extranets*[tea18a]. Within an extranet, several enterprises can collaborate as if they were in the same enterprise. There can be direct messages between the employees in the different enterprises and they are able to share knowledge with each other. Through the extranets, enterprises are able to restrict access to information that is only relevant within the collaboration[tea18a]. The tool provides collaboration opportunities such chats, white-boards and document sharing as well. A general name for these technologies are e-learning[LRSS03].

Figure 15 shows a possible setup for a system including Microsoft Sharepoint. An enterprise has a Sharepoint platform, where it is connected to its collaborators through the extranets created. Through the extranets, they are able to give the collaborators restricted access to only files relevant to them among other collaboration possibilities.

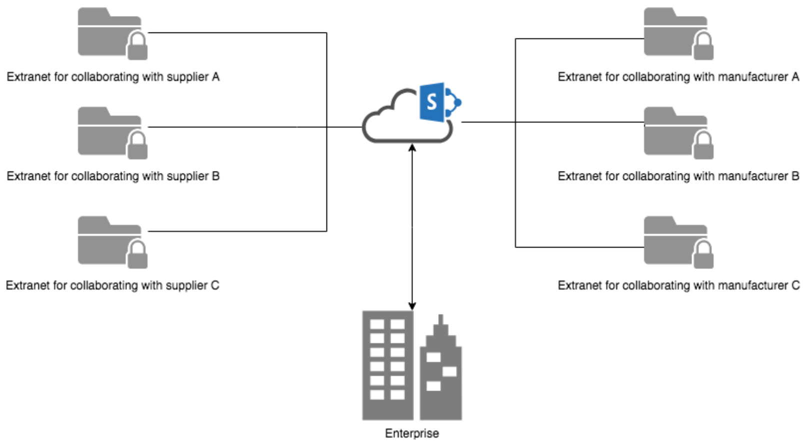


Figure 15: Microsoft Sharepoint architecture

19.3.2 Azure AD B2B

Azure AD B2B is a tool from Microsoft that provides business-to-business collaboration capabilities. The tool was launched during 2017, and already have more than 8 million individual users[Sim18]. Azure AD makes it easy for enterprises to work across borders and share knowledge and information with other enterprises, no matter the size[Woo18b]. Through the tool, enterprises are able to give access to documents and other files to their enterprise partners[Woo18a]. Even when providing access, the enterprise itself can maintain absolute control over their own resources.

Azure AD B2B is designed to work with any partner, even though they might not have Azure AD B2B themselves[Woo18b]. Collaborating with either Azure AD users or non-users, both are easy and straight-forward. Enabling connections are invitation-based and users that do not have Azure AD already will be given the opportunity to create an account.

As for Microsoft Sharepoint, Azure AD B2B also provides chat opportunities between enterprise partners which can make collaboration and knowledge and information sharing easier[Sta17].

19.4 Summary

This chapter has presented approaches and technologies that are relevant for the capability extroversion. Approaches that were defined were open innovation, co-creation and social technologies. Two social technologies were presented that could be used for enhancing extroversion, which were Microsoft Sharepoint and Azure AD B2B. These are both technologies that can be used for team collaboration cross-enterprise. For an enterprise to succeed with extroversion their is a need for a willingness to collaborate and open enterprise borders.

20 Summary

Table 21 shows the different approaches and technologies that are identified for the different capabilities within the framework. The enhanced framework will be presented in section 28.1.

Capability	**Approaches**	**Technologies**
Decentralization	• Spread decision making • Knowledge sharing • Setting restrictions for departments	• Slack • Skype for business • Microsoft teams
Interoperability	• Communication protocols • Standardization • Integrated approach • Unified approach • Federated approach • Use of broker	• IDEAS • Athena Interoperability Framework
Awareness	• SWOT • SCOPE • SOAR • Data analysis	• Sensor data
Perceptivity	• Historical data analysis • Real-time data analysis	• Apache Spark • Hadoop Map/Reduce • Amazon Elastic MapReduce
Prediction	• Machine learning • Complex event processing	• Amazon Sagemaker • Apache Spark • Oracle CEP • StreamBase
Intelligence	• Decision support system • Intelligent DSS • Model-driven DSS • Data-driven DSS	• Amazon Sagemaker • Apache Spark • Oracle CEP • StreamBase
Action	• Human interaction • Automated processes	
Extroversion	• Social technologies • Open innovation • Co-creation	• Microsoft Sharepoint • Azure AD B2B

Table 21: Approaches and technologies for each framework capability

Part VI
Prototype development - Predictive maintenance system

This part will present the development process and evaluation of the prototype developed for Business A. The prototype is developed for being used within predictive maintenance on their enterprise machines. The prototype will be a prediction model that will be able to classify whether an enterprise machine is likely to fail within the near future.
The enterprise will be presented, and an evaluation of them using the *framework for proactive enterprises* prior to the prototype will be done. A description of the problem at hand and why it can be vital to them, as well as the data received from them, will be presented. Possible architecture and technologies are shown in this part, and the development process is explained. The part is finalized by giving a proposed solution and a chosen algorithm.

21 Enterprise presentation

The enterprise, *Business A*, that collaborated with the author provided the data used for the machine learning is within the recycling industry. Specifically, the enterprise develops, manufactures and sells reverse vending machines (RVM). RVM's are machines that accept bottles and cans to be returned after use by customers who have bought them. When customers return the items, they receive money or some other kind of currency they can use to buy products. RVM's are typically stored in supermarkets and grocery stores and are easily accessible for customers. Due to the signing of a non-disclosure agreement, the enterprise will be kept anonymous throughout the thesis. Therefore, there will be no further presentation of their number of machines.

22 Evaluation of Business A prior to work with graph from research project

This chapter summarizes the results from the research project, where Business A was classified as a *liquid enterprise* by evaluating its proactivity level using the *framework for proactive enterprises*.

NOTE! This section was also included in the research project report

Business A is an enterprise with departments in several countries in the world. It develops and manufactures products for the recycling industry.

An interview was held with a worker from one of the departments which could act as a spokesperson for the enterprise.
For the interview, an enterprise evaluation form was used, which can be seen in appendix A. The interview was approximately 45 minutes. The interview was done Wednesday 29.11.2017.

The interviewee said that Business A was run with a top-down approach with several layers of management. Business A has different departments that work in specific areas, but the enterprise does not have a flat hierarchy. Decentralization is present within their machines as different components have different responsibilities within the machines. Since Business A has a top-down approach, the enterprise does not score better than the value 4 for decentralization.
As for interoperability, the interviewee told that they have several systems which can collaborate and act together. Calculating transactions was one of the tasks the systems collaborated for. As seen later in this section, Business A collaborates with other enterprises to deliver products. With their systems, the enterprise scores 12 for interoperability.
Business A does have a presence of sensors monitoring the state of the different components within their machines. These sensors are used for generating status reports which are sent to the back office. The enterprise is also doing environmental awareness through governmental affairs and searching for new market opportunities. Due to this fact, the enterprise scores a 12 in awareness based on the *framework for proactive enterprises*.
Through these sensors and status reports, Business A gathers enormous amounts of data. This data is analyzed and made meaning of, and are to be used. This gives the enterprise a score of 12 in perceptivity, while they get a 0 in prediction. There was no or basic prediction done by systems which resulted in the score given for prediction.
When it comes to the capability of intelligence, the enterprise scores a 4 in the *framework for proactive enterprises*. This is reasoned by the fact that they do not utilize intelligent decision-making systems for optimized decision making. They do, however, discover, store and analyze behavior patterns.
For the most part, actions are taken reactively. However, for some parts of the product, a report is generated and given to workers about statuses. By doing this, the workers can act proactive and change components before breakdown. Since actions are taken proactively by humans, Business A scores 6 for the capability of action.

The enterprise does collaborate with other enterprises. The interviewee could tell a specific case where the enterprise collaborated with another enterprise to deliver a product where both enterprises contribute with expertise and information. Due to the collaboration, the enterprise scores an 8 for the capability of extroversion.

Based on the interview, the enterprise ended up with the values specified in table 22 and the graph in figure 16.

Capability	**Value**
Decentralization	4
Interoperability	12
Awareness	12
Perceptivity	12
Prediction	0
Intelligence	4
Action	6
Extroversion	8

Table 22: Values for Business A during the development of the framework

Looking at the combinations specified in the combination box beneath the radar chart, we see that Business A has several weakness areas. Due to the lack of decentralization, prediction, intelligence and action we identify weakness areas to be proactivity, intelligence, the sensing capability and the enterprise are not classified as collaborative. Because the enterprise gets full scores on awareness and perceptivity, we can classify a strength as the enterprise being aware.

Comparing the results Business A got with the future enterprise types specified in the *framework for proactive enterprises*, the enterprise was placed between two enterprise types. These enterprise types are explained in chapter 5. This can be seen in figure 17. The two future enterprise types were liquid enterprise and partially proactive enterprise. Due to the lack of prediction and intelligence, the enterprise could not be classified as a partially proactive enterprise.
Looking at the results Business A got, the framework suggests that an intelligent decision-making system can be integrated into the system to enable intelligence and prediction. Business A is already innovative and agile to some degree, but to further advance to a proactive enterprise, they need to reorganize and have a flat hierarchy to enable faster innovation and better agility.

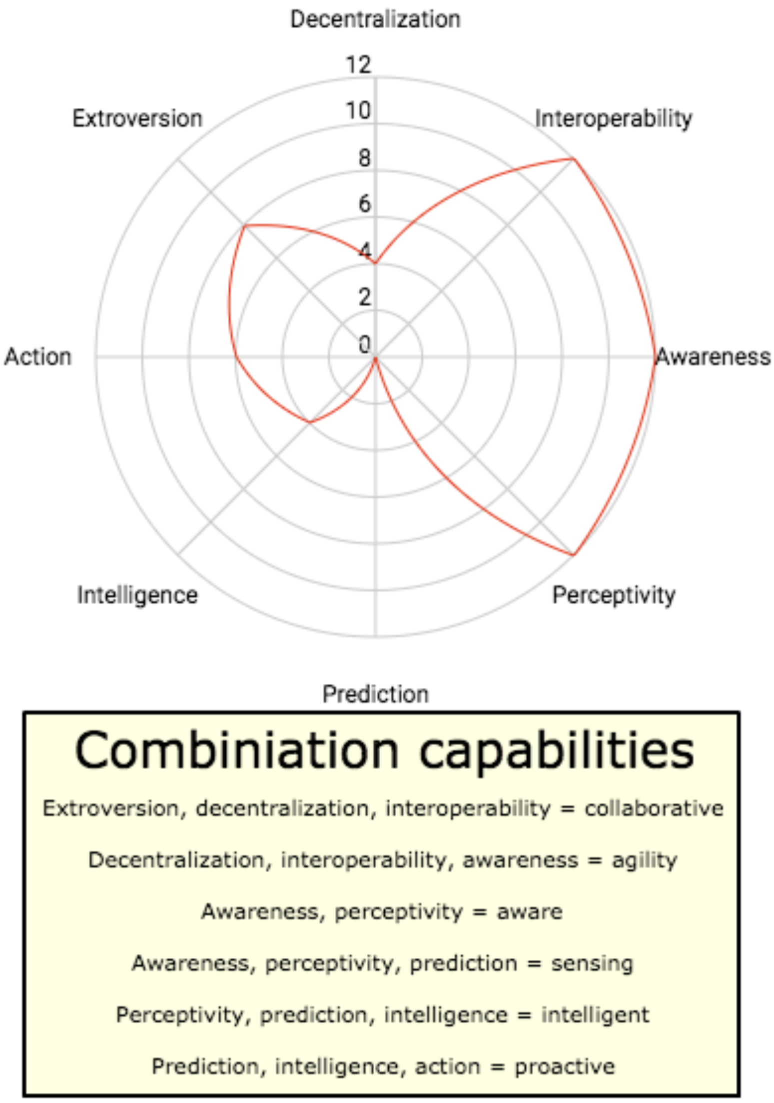


Figure 16: Results for Business A

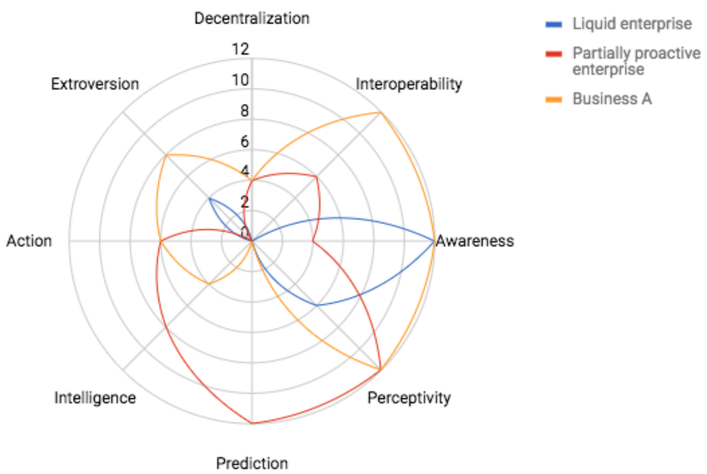


Figure 17: Comparison of Business A

23 Problem description

To get a greater understanding of the importance of a prototype for predictive maintenance for the enterprise, the author had a structured interview with Business A. The template for the interview can be found in appendix C. Some of the questions they could not answer as they were too sensitive for the enterprise. Through the interview, it was also possible to map the value chain, which will be explained in the following paragraphs.

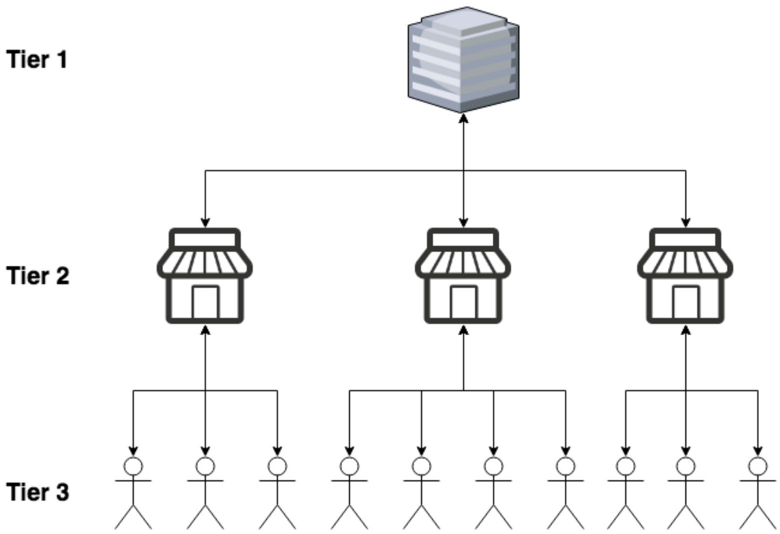


Figure 18: Value chain

Figure 18 shows the enterprise value chain. The enterprise (tier 1) is manufacturing and maintaining RVM's which they sell or rent out to their customers (tier 2). The customers of the enterprise are normally normal supermarkets, gas stations and similar. These are the direct customers to the enterprise. Naturally, the supermarkets and the other customers of the enterprise have customers themselves (tier 3). The customers of the supermarkets etc are normal humans living the everyday life. The prototype that will be developed will be inserted into tier 1, and will be a part of the back office that is within the enterprise. The RVM's are located at tier 2, and the prototype will directly affect tier 2. The prototype will indirectly affect tier 3 through tier 2.
If a machine at tier 2 is out of service, that leads to customers at tier 3 begin unsatisfied, as they cannot use the machine at the current store they are at. An unsatisfied customer at tier 3 can lead to the customer going to another store to be able to use the machine there and also do their groceries there. This might also lead to the customer choosing to do this in the future as well.
Another consequence of a machine out of service at tier 2 is increased maintenance costs and decreased income for the enterprise at tier 1. For the customers at tier 2, loosing customers and therefore loosing income might lead to unsatisfaction with the enterprise at tier 1. An unsatisfied customer at tier 2, a store, will result in the store to choose one of the enterprise competitors in the future when they are investing in RVM's. If customers at tier 2 chose other suppliers, it results in the enterprise loosing customers and income.

Questions were asked of what type of maintenance Business A was conducting as of today, where they told that reactive, preventive and predictive maintenance all were used, but that maintenance mostly was done reactively. That opens doors for improvement and a need for predictive maintenance. To conduct predictive maintenance, much data is needed to be analyzed. The enterprise told that each machine produces 1-3 MB of data each day and that status reports are sent from the machines every third minute at least.

Due to enterprise sensitivity some questions about how often maintenance was required and how far ahead it had to be planned, how much it cost and how much of the yearly budget it consumed could not be answered.

23.1 Problem definition

The problem definition comes from the problem description. Several severe consequences are presented, which can be addressed by using predictive maintenance. Predictive maintenance can reduce the degree of the consequences by reducing downtime on the enterprise machines, and the affect it can have on the value chain are presented in section 26.1. The problem at hand is a binary classification supervised learning problem. It is a supervised learning problem as the desired outcome for each data row are present during training. The two classes which we are using are "running" and "not running," which indicates whether a machine is in or out of service. The problem has the following problem definition:

Use the machine learning workflow to process and transform data from generated RVM status reports to create a prediction model. This model should predict likely RVM failure in the near future with 90% or greater accuracy.

23.2 Dataset description

Business A provided data generated by their RVMs. In total, almost 300 GB was received from them which was uploaded to an S3 bucket on an AWS account. The 300 GB of data was divided into status messages sent by the machines to the enterprise back office where they are processed. There were seven different kinds of status messages, which was reporting different things. The following sections will explain the status messages further, but the information that can be provided is limited, and message names and similar have been censored. Each of the status reports will instead go by the abbreviation SR. All of the report types have timestamps and can be connected to a particular machine which is of use to connect the different report types and gather them to describe the state of a machine at a given time. Table 23 shows an overview of the status report types with a short description.

SR	Description
SR1	Information about RVM bin states
SR2	Information about customer session data
SR3	Lets back office know machine is still running
SR4	Information about RVM state
SR5	Includes notification messages about RVM state
SR6	Information about machine configurations
SR7	Information about component states monitored by sensors

Table 23: Business A dataset description

23.2.1 SR1

Each type of RVM has a set of bins where items inserted are placed. It can be either bottles or cans, where the items are first compressed, so they occupy less space in the bins. The bins need to be emptied by store personnel for the machine to be able to function properly.
The machines typically have bins that are asserted to either bottles or cans. This assertion results in the machines needing to be emptied at a point where either the bins for cans or bottles are full. The store personnel is advised when the bins are full, and the store customer is not allowed to place any more items in the machine until at least one of the bins are emptied. If all bins, for either bottles or cans, are full for a machine at any time, that results in downtime for the machine and thereby unhappy customers that have to wait. The first status report type is concerned with the statuses of each of the bins in a certain machine. The status report holds the status of each

bin, as well as the estimated level within the bin. Another variable in this report counts all the items that have been sent to a certain bin since the last time the machine was reset.

From this report, the bin levels and statuses can be extracted which can help predict full bins before they go full. Being aware of that a bin (or all of them) might be full shortly makes it possible for a store employee to schedule the task better and empty it when he is idle.

23.2.2 SR2

The time span from when a store customer enters the first bottle or information until he receives the payment is called a session. The second status report type is about sessions. One report of this type corresponds to one session, with information about when it started, when it ended, the amount of value the customer got in total, as well as which method was used for payment. Each report also includes a list of items that are inserted into the machine during the session. A lot of information about each item is also presented, such as what type of item it is, the weight of the item and so on.

From this report type, extraction of the total count of sessions for a machine, the total time of session span and the total count of each of the payment methods can be done. All of these variables can wear out machine parts and are therefore essential to include in the data analysis.

23.2.3 SR3

The third status report type does not hold much additional information compared to the other status types; in fact, it does not hold any information that is not present in the other status report types. The report is a heartbeat report, and the point of it is just to let the back office know that the machine is up and running and connected to the internet. This report is sent from the machine to the back office at least every third minute, and it is mainly sent if none of the other report types are sent.

This report type can be used to know that the machine is up and running even when the other status report types are not sent. However, if a machine is not connected to internet, this status report type does not get buffered.

23.2.4 SR4

The fourth report type is about the state of a machine. It clarifies whether a machine is running or if it is down. If the machine is down, the report also presents a reason out of eight possible reasons alongside an error message which explains why the machine is not running.

This report type is crucial for the data analysis as it contains real cases of when the machine was not working and why it was not working. Furthermore, a machine will have the same state until the next status report of this type is sent. This helps in the data handling as assumptions can be made that the state will be the same until a new report has arrived.

23.2.5 SR5

A machine consists of several components that are put together to work together but have different responsibilities when it comes to the functionality of the machine. The fifth report type sends notification messages where severity is defined out of five possible. To further explain the notification, a message is included which reasons the severity type given and what the problem is. The report can also include which component is the one that is affected.

This report can be used in the data analysis as it, as the previous report, contains real cases of when the machine was not working and why it was not working.

23.2.6 SR6

The sixth report type holds information about machine configurations and will not be included in the data analysis. This is reasoned as such because the data analysis is concerned with physical machine state and not software state which can introduce bugs. It is also left out of the data analysis due to sensitive information that this report type holds.

23.2.7 SR7

The seventh and last report type presents information about component states monitored by sensors. Example of information provided is machine CPU temperature, dirt level on different components and so on. Another example is free disk space on the machine.

This report type holds a lot of information that can be of value for the data analysis. Variables such as CPU temperature can be crucial for predicting machine failure and will, therefore, be included in the data analysis.

23.2.8 Dataset summary

To be able to use all of this information in the data analysis, the report types need to be connected into one by using the information about which machine they belong to and the timespan they were sent in. The report types present a lot of valuable information in total, but there is also a lot of information which are not useful for the data analysis that will be removed from the final data that will be used for training the prediction model.

23.3 Description of scenario

The imagined structure for the enterprise when the prediction model can be used can be seen in figure 19. Each RVM machine generates the seven status reports as described in section 23.2. The status reports are sent to the enterprise back office where they are connected and gathered into one report. This report is then sent to the trained prediction model which is hosted on an AWS Sagemaker endpoint (Explained further in section 16.3.1). Then the model analyses the new set of data and returns a response whether the machine is likely to fail soon or not. Based on the response, actions can be taken.

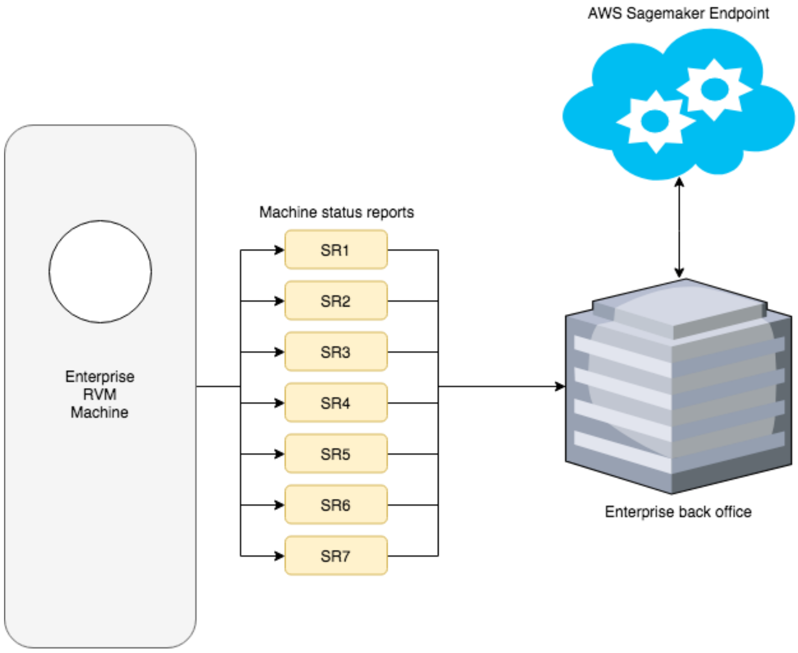


Figure 19: Enterprise scenario

24 Technologies

In this chapter, the technologies that are used during development of the predictive model for the predictive maintenance for the RVMs are presented.

24.1 Amazon Sagemaker

A presentation of Amazon Sagemaker was given in section 16.3.1. In this case, access to generated RVMs status reports was obtained, as described, on an AWS S3 bucket which is loaded into a Jupyter Notebook instance on Amazon Sagemaker.

24.2 Jupyter Notebook

Jupyter Notebook is an open-source web application that lets a developer create and share notebooks that can contain live code and narrative text. It can also contain equations and visualizations. It is handy to use as the outcome of the code and describing text look very good. Another useful functionality of Jupyter Notebook is that it can run a program partially and the developer decide which parts to run.

When working on Amazon Sagemaker, as described in section 16.3.1, the machine learning workflow is done through Jupyter Notebooks in the cloud.

25 Development process

In this chapter, it will be described the different steps of the development process for the predictive maintenance prototype developed. The different steps are data selection, data preprocessing, data cleaning, selecting the machine learning algorithm, training the model and testing the model.

The Jupyter notebook instance used during the development process is based on the tutorial from Amazon called *Train faster, more flexible models with Amazon SageMaker Linear Learner*[GVCK18].

25.1 Data selection

As presented in section 23.2, the author received 7 different types of status reports. All of them containing information about different aspects of an RVM. Not all of the information was relevant, and the author needed to go through and study the different report types to understand the data they presented. Iterations were done to figure out what information might be useful.

25.1.1 Criteria

Table 24 presents the criteria used for data selection.

Criteria Number	Criteria
C1	The data presents a physical value of an RVM
C2	The data is not duplicated or could be found in another report type
C3	The data can be turned into numerical values

Table 24: Data selection criteria

25.1.2 Results

The initial plan was to use all the seven status report types to construct the data to be used, but after several iterations, the author discovered that some of them were not useful. SR2 was completely removed as it did not fulfill criterion C2 from table 24. SR6 contained information about machine configurations and therefore did not fulfill criterion C1 from table 24. The author decided that it would not be included because the relevant failures to predict are due to physical values and not software bugs. A lot of the status report types held messages or information in the string about what had happened. The author concluded that they would not be included as they went against criterion C3 from table 24. They could have been turned into numerical values, but the number of messages was so large, and the author identified the severity and the reason for the message as more important than the extra information the messages provided.

This left five status report types that should be included; SR1, SR3, SR4, SR5, and SR7. The initial 300GB that was received from Business A was reduced to 76,5 GB after the data selection.

25.2 Data preprocessing

As mentioned, the data was retrieved through an S3 bucket where Business A uploaded all the data. Since the data came in several different folders and files and for different status report types, there was a necessity to make a program which would iterate through the files and find useful information. The program would also need to connect the different status report types that belonged to the same RVM (C5 in table 25) at a given point in time (C6 in table 25). A target value for the machine learning was included based on the values analyzed to say whether the machine was running or not. The program flow is described in appendix D.

All the data ran through the program twice, where the two rounds had different criteria. The data that the program ran through was from the period of August 2017 until March 2018.
The first round had the criteria C4.1, C5, and C6 from table 25. The program ran through all the data and generated CSV files based on the data and the criteria provided.
The second round had the criteria C4.2, C5, and C6 from table 25. The program ran through all the data and generated CSV files based on the data and the criteria provided.
Each of the rounds had an output of a generated CSV file which contained all the filtered and prepared data.

25.2.1 Criteria

Table 25 presents the criteria used for data preprocessing.

Criteria Number	Criteria
C4.1	5/5 status report types were present
C4.2	4/5 status report types were present, SR3 can be missing as it is mostly sent when the others are not
C5	All values came from the same machine
C6	All values were sent from the machine within the same minute

Table 25: Data preprocessing criteria

25.2.2 Results

It took a while to process the data. For the first round, the program ran for almost 12 hours. The second round had looser criteria which ended up generating more data, and it ran for 18 hours.

The data preprocessing filtered away a lot of data. There was stricter criteria for round 1, which ended up generating a CSV file containing 65580 rows, and each row had 57 variables. The CSV file had a size of 17,5 MB. For round two the generated CSV file contained 124772 rows, and each row had 57 variables. The size of the CSV file was 34,9 MB.
As we can see, the second round ended up generating an amount of data which was almost twice as big as the first round.

25.3 Data cleaning

When the generated CSV files were ready, they were uploaded to the same S3 bucket as used earlier. From there it was possible to import them to a Jupyter Notebook instance at Amazon Web Services through Amazon Sagemaker. The Jupyter Notebook can be seen in appendix E. In the notebook instance, more data preparation and cleaning was done.
First of all, the columns which were not about machine status (C7 from table 26), but rather about machine information, were removed. This reduced the number of columns from the original 57 to 54.
Secondly, all rows that contained only missing values (C8 from table 26) were removed. There should be none of those, but likewise, it was done to be sure.
Thirdly, all columns that contained more than 50% missing values were removed from the dataset (C9 from table 26). This reduced the number of columns from 54 to 12.
Lastly, the remaining missing values were replaced with 0's, as the machine learning algorithm require numerical values (C10 and C11 from table 26). To be sure, all column data types were also converted to float32.

25.3.1 Criteria

Table 26 presents the criteria used for data cleaning.

Criteria Number	Criteria
C7	Column must be about machine/component state or status
C8	A row cannot contain only missing values
C9	Column cannot have more than 50% missing values
C10	A row cannot contain missing values
C11	A row can only contain numerical values

Table 26: Data cleaning criteria

25.3.2 Results

The data cleaning removed many columns and a lot of the data with it. The initial 57 columns were reduced down to 12 as many columns contained more than 50% missing values.

Data exploration gave me the the results presented in table 27 and table 28. As we can see, most of the data was not running instances, and the datasets were very imbalanced. The strict criteria C4.1 and C4.2 from table 25 made the dataset so imbalanced as the requirement of all, or nearly all status report types required the status report types that reports failure to be present.

Value	**Amount**	**% of total**
Not running	55781	85,1%
Running	9799	14,9%
Total	65580	100%

Table 27: Data exploration results for round 1

Value	**Amount**	**% of total**
Not running	102801	82,4%
Running	21971	17,6%
Total	124772	100%

Table 28: Data exploration results for round 2

25.4 Selecting the machine learning algorithm

Several factors need to be considered when choosing a machine learning algorithm. The first and most important factor is to understand a problem and to define what is going to be the input type and what is going to be the output type. The author defined the problem and decided the input and output type in section 23.1. The definition lead to the criterion that the algorithm must handle supervised learning (C12 from table 29) and the criteria that the algorithm must handle binary classification (C13 from table 29). The reason why it is supervised learning is because the desired value is known before the training is conducted. As section 25.3.2 presented, our data is highly imbalanced, which lead to the introduction of the criterion that the algorithm must be able to handle imbalanced classes (C14 from table 29). Since the author was not very familiar with machine learning prior to the thesis work, a criterion was introduced that the algorithm must be within the set of built-in algorithms at Amazon Sagemaker (C15 from table 29). This criterion would make it possible to train the prediction model without too much knowledge about machine learning.

25.4.1 Criteria

Table 29 presents the criteria used for selecting the machine learning algorithm.

Criteria Number	**Criteria**
C12	The algorithm must be fit to handle supervised learning
C13	The algorithm must be fit to handle binary classification
C14	The algorithm must be able to handle imbalanced classes
C15	The algorithm must be within the set of built-in algorithms at Amazon Sagemaker

Table 29: ML algorithm selection criteria

25.4.2 Results

That the algorithm must be included in the set of built-in algorithms at Amazon Sagemaker (C15) left me with about 11 different algorithms to choose from:

- Linear learner
- Factorization machines
- XGBoost algorithm
- Image classification algorithm
- Amazon SageMaker Sequence2Sequence
- K-Means algorithm
- Principal Component Analysis (PCA)
- Latent Dirichlet Allocation (LDA)
- Neural Topic Model (NTM)
- DeepAR Forecasting
- BlazingText

Including the requirement for the algorithm to handle supervised learning (C12) reduced the set to 5 different algorithms:

- Linear learner
- Factorization machines
- XGBoost algorithm
- Image classification algorithm
- Amazon SageMaker Sequence2Sequence

Including the requirement for the algorithm to be suitable for binary classification (C13) reduced the set to three algorithms:

- Linear learner
- Factorization machines
- XGBoost algorithm

Including the criterion for the algorithm to handle the imbalanced data in the dataset left two possible algorithms:

- Linear learner
- XGBoost algorithm

The two different algorithms handle resolving of imbalanced data in two different ways. The linear learner automatically calculates the weight value to be used and is, therefore, the preferred algorithm.

The chosen algorithm for training the prediction model is, therefore, the *linear learner algorithm*[13] at Amazon Sagemaker.

[13] https://docs.aws.amazon.com/sagemaker/latest/dg/linear-learner.html

25.5 Training model

To train the model, the first thing done was to read the data and then shuffle it to have a random order of the data. The next step done was to split the data into test and training set, where the labels (the actual state of the machine at the time) were separated from the features used.

When training on Amazon Sagemaker, the user must define an instance type that is going to be used to run the actual training. In this case, an ml.m4.xlarge instance was used which has 8GB RAM and a bandwidth of 450 Mbps. It is suitable for small and medium-sized databases and therefore fit for the datasets obtained.

When running the training process, there must be defined some hyperparameters for the algorithm which can help a developer tune the algorithm as he or she wants. For the algorithm used in this case, the linear learner, there are only two hyperparameters which are required; those are *feature_dim* and *predictor_type*. *feature_dim* requires the number of features included in the set, while *predictor_type* requires the type of output which we want which could be either binary classification or regression. In this case the *predictor_type* is *binary_classifier*.

The training was run twice, where one of the models included a hyperparameter to balance the dataset that we have (logistic with class weights). The other approach did not specify that anything needed to be done to balance out the classes (logistic). Appendix E presents programmatically how the training set-up was done.

25.6 Testing model

During the training process, explained in section 25.5, HTTP endpoints for each model trained are created. These endpoints work such as they receive the same features specified as during training and they will come with a prediction based on the previous training. Based on the results, calculations are done for recall, precision, accuracy and F1 score as described in section 8.2.3.

The actual testing of the models is done by sending the test set to the prediction models, where the result of the prediction is compared to the known value. The results of the testing can be found in section 27.1.

26 Proposed solution

In this chapter, the author will present the proposed solution for the predictive maintenance developed for the enterprise and the effects it may have on the value chain. The results obtained by using the chosen algorithm on the data possessed gave the results presented in chapter 27.

26.1 Effects on value chain

The value chain for this certain case was presented in chapter 23. If the prototype can be integrated successfully into Business A, then it will affect all three of the tiers presented.
A successful integration of predictive maintenance will result in reduced downtime for the enterprise machines if Business A acts proactively. Reduced downtime of the enterprise machines ultimately results in increased customer satisfaction. At tier 3 of the value chain, the customer satisfaction will be increased by the increased availability of the machines to the customers. They will be able to recycle their bottles and cans at the store without going to another place to do it. At tier 2, customer satisfaction will be increased as the customers at tier 3 will stay loyal to their store. Loyal customers results in increased income for the stores at tier 2. As customer satisfaction at tier 2 is increased, income will increase for Business A at tier 1. Predictive maintenance can also help Business A reduce maintenance costs.

26.2 Chosen algorithm

As mentioned earlier the chosen machine learning algorithm ended up to be the linear learner algorithm through Amazon Sagemaker. The criteria for why it was chosen can be found in section 25.4.1.

The linear learner algorithm is a supervised learner algorithm and can be used for both regression and binary classification problems. It expects a data matrix where each row contains the values for each feature, and each column represents a feature in the data. An additional column is required for the target value, which for binary classification problems are expected to be either 0 or 1.
The algorithm also has possibilities to handle imbalanced classes in the data and was one of the reasons for why it was chosen. Previously, imbalanced classes could be a massive problem for machine learning, or at least cause problems and had to be handled. The linear learner transforms this problem into a problem that is highly possible to overcome.

Part VII
Results

This part of the master thesis will present the results obtained during the project. It will consist of two different chapters, where chapter 27 presents the results for the prototype, while chapter 28 presents the enhanced framework. Chapter 27 presents the results obtained through the different rounds testing the prediction model developed. The test set was used during testing. This chapter will also include the results for the new evaluation of Business A obtained through the framework after introduction of the prediction model at the enterprise. As mentioned, chapter 28 will present the enhanced framework as well as summarize the approaches and technologies found for each capability.

27 Prototype results

This chapter will present the results obtained when running the prediction model, while discussions about the prototype will be done in section 29.1.

27.1 Results

In this section, the author will present the results gotten for the prototype testing as described in section 25.6. The different metrics used in the different columns in table 30 and 31 are explained in section 8.2.3.

27.1.1 Round 1

The first round of the machine learning program was run with the smaller amount of data as presented in table 27.

The results are presented in table 30. It can be seen that the prediction model has high values for each of the evaluation metrics specified. Discussions will be made in part VIII.

	Model	**Recall**	**Precision**	**Accuracy**	**F1**
0	Logistic	0,991	0,991	0,997	0,991
1	Logistic with class weights	0,921	0,999	0,988	0,959

Table 30: Prediction model results for round 1

27.1.2 Round 2

The second round of the machine learning program was run with the more significant amount of data as presented in table 28.

The results are presented in table 31. It can be seen that the prediction model has high values for each of the evaluation metrics specified. Discussions will be made in part VIII.

	Model	**Recall**	**Precision**	**Accuracy**	**F1**
0	Logistic	1	0,982	0,997	0,991
1	Logistic with class weights	0,951	1	0,991	0,975

Table 31: Prediction model results for round 2

27.2 Evaluation of Business A post introduction of predictive maintenance

In this chapter, the author will use the *framework for proactive enterprises* to evaluate the enterprise for which the prediction model was developed. The previous evaluation, presented in section 22, can be found in chapter 22. The evaluation assumes that the prototype, the prediction model developed, is successful and integrated into the enterprise. The architecture proposed in section 23.3 can be used to integrate the prediction model.

Based on the previous values, presented in chapter 22, and the updated values, presented in this section, for prediction and intelligence based on the introduction of the prediction model, the enterprise ends up with the values specified in table 32 and the graph in figure 20. The values are extracted by using the tables presented with the *framework for proactive enterprises* presented in part II.

Capability	Value
Decentralization	4
Interoperability	12
Awareness	12
Perceptivity	12
Prediction	12
Intelligence	8
Action	6
Extroversion	8

Table 32: Values for Business A during the development of the framework

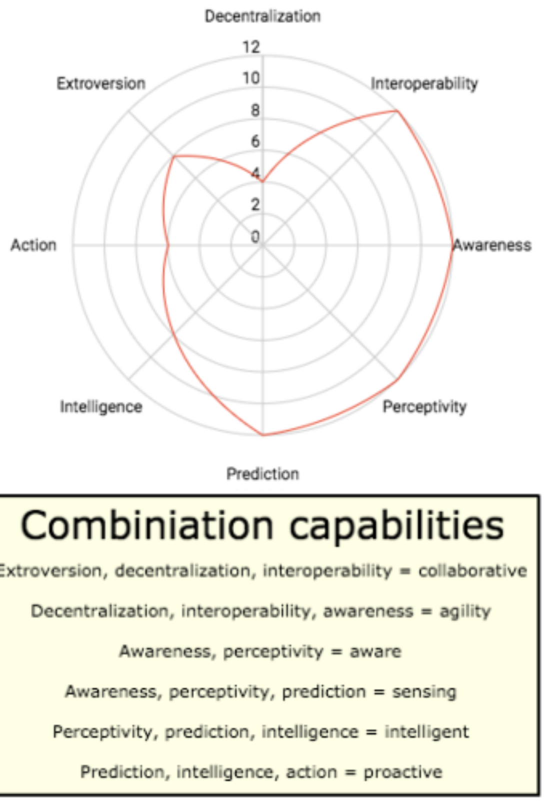


Figure 20: Results for Business A post introduction of predictive maintenance

Comparing the new table and graph, we can see that the improved values are within the capabilities prediction and intelligence. Earlier there were defined weakness areas for the enterprise to be proactivity, intelligence and the enterprise is not classified as collaborative. Because business A got full scores on awareness and perceptivity, we could classify a strength as the enterprise being aware. With the new values, the enterprise has transferred some of the weaknesses into being strengths instead. The strengths of the enterprise can now be set as proactivity, intelligence and being aware, while a weakness that remains is the lack of collaboration possibility.

Earlier, when comparing the results Business A got with the future enterprise types specified in the *framework for proactive enterprises*, the enterprise was placed between two enterprise types. These enterprise types are explained in section 5. The two future enterprise types were liquid enterprise and partially proactive enterprise. Due to the lack of prediction and intelligence, the enterprise could not be classified as a partially proactive enterprise. Suggestions were made to develop a predictive model to enable intelligence and prediction which is now done.

Now, when comparing the results Business A got with the future enterprise types specified in the *framework for proactive enterprises*, we see that the enterprise has evolved and now surpassed the partially proactive enterprise. Therefore the framework now defines the enterprise as a partially proactive enterprise. The results can be seen in figure 21.
Suggestions for areas to improve are the collaboration possibilities and open enterprise boundaries to spark collaboration possibilities.

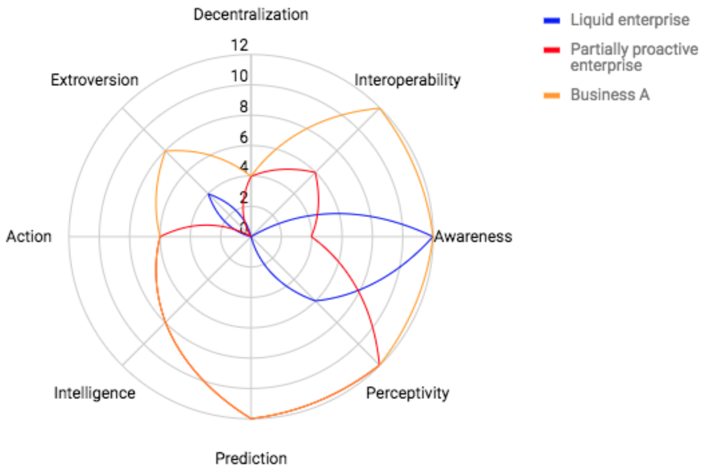


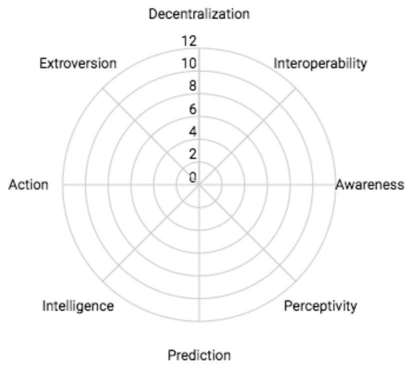
Figure 21: Comparison of Business A post introduction of predictive maintenance

28 Proposed enhancement of framework

Based on the literature review done and the results gotten, an enhancement of the *framework for proactive enterprises* can be done. The basics of the framework itself have not changed too much, but the enhancement consists of providing guidelines for enterprises on which approaches and technologies can be used within the different capabilities to advance as an enterprise. In the framework, the approaches and technologies are included, but not presented, as an enterprise can find further information about the guidelines in part V of this master thesis.

28.1 Framework enhancement

The enhanced framework will consist of the framework as it used to be, but including approaches and technologies that can be used within the different capabilities.
Figure 22 shows the proposed solution for the enhanced framework.



Combiniation capabilities

Extroversion, decentralization, interoperability = collaborative

Decentralization, interoperability, awareness = agility

Awareness, perceptivity = aware

Awareness, perceptivity, prediction = sensing

Perceptivity, prediction, intelligence = intelligent

Prediction, intelligence, action = proactive

Capability	Approaches	Technologies
Decentralization	• Spread decision making • Knowledge sharing • Setting restrictions for departments	• Slack • Skype for business • Microsoft teams
Interoperability	• Communication protocols • Standardization • Integrated approach • Unified approach • Federated approach • Use of broker	• IDEAS • Athena Interoperability Framework
Awareness	• SWOT • SCOPE • SOAR • Data analysis	• Sensor data
Perceptivity	• Historical data analysis • Real-time data analysis	• Apache Spark • Hadoop Map/Reduce • Amazon Elastic MapReduce
Prediction	• Machine learning • Complex event processing	• Amazon Sagemaker • Apache Spark • Oracle CEP • StreamBase
Intelligence	• Decision support system • Intelligent DSS • Model-driven DSS • Data-driven DSS	• Amazon Sagemaker • Apache Spark • Oracle CEP • StreamBase
Action	• Human interaction • Automated processes	
Extroversion	• Social technologies • Open innovation • Co-creation	• Microsoft Sharepoint • Azure AD B2B

Figure 22: The enhanced framework

By including the approaches and technologies for each of the capabilities, the framework was enhanced. The results from the literature review provides an answer to the question *how* an enterprise can advance within each capability. By specifying approaches and technologies, it is now easier for an enterprise to begin their transformation towards a proactive enterprise. Earlier, the framework did not answer the question of *how* to improve within each capability, but that question is now answered.

An example of a use case is that an enterprise wants to improve within the prediction capability. The enterprise would quickly be able to see that approaches that can be taken are machine learning and complex event processing.

Part VIII
Discussion

In this part of the thesis, the results achieved for both the prototype and the enhancement of the framework will be discussed. Discussions and evaluations for the research methodology will be done as well as to see how well the research questions were answered. The fulfillment of the project goals will also be discussed.

29 Result analysis

This chapter will discuss the results achieved for the prototype and the results achieved from the enhancement of *framework for proactive enterprises*.

29.1 Prototype evaluation

As we could see in chapter 27, the results for the prediction model were very good. They exceeded the expectations of at least an accuracy of 90% as defined in the problem definition in section 23.1. It is necessary to try to understand why the results were so good, and if they, in fact, are the best results possible to get.

29.1.1 Data evaluation

A lot of data was received for this task, which is good as machine learning training require a vast amount of data to avoid challenges such as overfitting. As mentioned, the data was cleaned and preprocessed before being used for training for the machine learning model. During this process, a target column was added to the processed data to have a known value for whether the machine was running or not at a given point. This value was decided based on the data given and communication with Business A. The fact that the values for the target column were decided in this way can affect the results of the prediction model, the reason being that the prediction model might suffer from overfitting and look for specific values which make for a specific outcome.

The data used had a lot of discrete values and not so many continuous values. This can lead to overfitting as the machine learning algorithm might look for the presence of the values which makes for a particular outcome. Furthermore, since the preprocessed data had so much missing data, a lot of the data was removed from training. A better result would have been achieved if the data did not have as many missing values so more features would be included in the training. The fact that the preprocessed data in the end only contained 12 features makes the results from the machine learning weak. As many features as possible should be included.

29.1.2 Machine learning algorithm evaluation

The machine learning algorithm chosen gave good results based on the data that was inserted into it. Due to Amazon Sagemaker, it was easy to work with and configure the algorithm and prepare it for the training. The training did not take too long, and the algorithm worked fast. When the algorithm got inserted more data during training, it could produce better results than if it received fewer data.

When it comes to handling imbalanced classes in the data we see in chapter 27 that when the algorithm was run configured to handle imbalanced classes it produced better results for the metrics precision and F1. Accuracy and recall scored better when the imbalanced classes were not taken into consideration. Both ways of running the algorithm produced excellent results, which strengthens the suspicion that the results are suffering from overfitting.

29.1.3 Challenges

Since the author did not have much experience with machine learning before the prototype development, some challenges were met. Before starting to train a model the author needed to run through some tutorials to understand how to work with machine learning and the machine learning workflow. Neither had the author any experience with Amazon Sagemaker or Jupyter Notebook, so it was necessary to study these technologies also before starting the actual work for the prototype. The lack of experience with machine learning was a challenge faced, but using time and tutorials helped overcome the challenge.

As the Java program described in appendix D ran on the data, some problems were encountered because there was too much data for the computer to handle. First, the author tried to iterate through the data on a monthly basis, but this made the program terminate as Java's garbage collector ended up working almost all the time trying to free memory, making the program not proceeding. To overcome this problem, it was necessary to alter the program to run through and process the data based on which day the report was sent. This lead to the same problem, and again it was necessary to alter the program, this time to run through the data based on the hour it was sent. That finally fixed the problem and the program could process all of the data without problems.

As mentioned, a lot of the data worked with was removed during data cleaning, and this was a challenge that needed attention. When first running the cleaning and seeing that it removed so much of the data the author went back to the preprocessing step and tried to iterate backward in time for each machine to obtain values that were possible to assume would be the same. As mentioned in the previous challenge, it was necessary to iterate through the status reports on an hourly basis which limited how far back it was possible to go. This made a lot of the missing values not being replaced with real values. The author could only overcome this challenge to a certain point, but it did not help to keep more of the columns during data cleaning.

One of the challenges faced was that the data had very imbalanced classes, as can be seen in tables 27 and 28. The method used to overcome this challenge was to use the linear learner algorithm from Amazon Sagemaker, but as the results in chapter 27 shows, it did not give very different results. For that reason, suspicions are raised that the model suffers from overfitting.

This was not necessarily a challenge as many rows of data for the first round of running the machine learning algorithm were obtained as well, but as the author tried to obtain more data, he went back to the preprocessing step and loosened up some of the criteria he had set for his data. This produced a CSV which contained almost twice the amount of data rows as when running with the strict criteria.

29.1.4 Lessons learned and recommendations

Data cleaning and preprocessing takes a lot of time, and a developer should expect to return to these steps. During the work with the prototype, the author experienced that a lot of time for cleaning the data and for preprocessing was necessary. A part of the reason is that, to begin with, all of the data has to be understood to know what should be included in the final dataset. This means that extracting real value out of the data is time-consuming and it can be challenging. Another reason was that the amount of data was so significant that it was hard for the computer used to process it. A solution here could have been to download the data to a more powerful server and run the program there. An example could have been an EC2 instance at Amazon Web Services. Spend a lot of time for data cleaning and preprocessing and think through decisions. During literature review of machine learning, several sources said that a data scientist would probably use most time on these steps, and from experience obtained, the author would have to agree with them. It is also highly likely that a developer would need to return to these steps even after running training sometimes.

The author is pretty sure that better results could have been achieved if the target column was already included in the data received and that it did not have to be decided during the data preprocessing. This makes the target column be based on values found during data preprocessing, and this can lead to unwanted behavior. So a recommendation is, if possible, not decide the target column during data preprocessing.

Considering a lack of experience with machine learning, it is recommended to use Amazon Sagemaker. At least for the first project, as it is easy to use, well documented and the community is growing. The author had a lot of good experiences using it, as it is made for making machine learning more manageable for people who are not too experienced with it.

Recommendations are made, that before starting the data cleaning and preprocessing, that a developer knows data handling. In this case, a lot of time was saved since the author is a computer scientist and knew how to handle the data and to structure it.

How much data is needed for a robust machine learning model is a difficult question to answer. As mentioned several times, the more data obtained, the better it is. Try to include as much data as possible as it will help achieve better-trained models. This recommendation can be backed by the results presented in chapter 27. We can see that for the second round that the program was run; the results were better for almost every single metric. So, the more data included, the better results will be achieved. Having quality of the data is essential, as well as having quantity. It is crucial to avoid the curse of dimensionality[KM17] when gathering data for the training.

29.1.5 Extracting value from enterprise data

Extracting value from existing data is important to make meaning out of the data and for it to be useful for an enterprise. For value to be extracted from existing data, it is important to review the data to fully understand what the data consists of and what kind of information it holds. Machine learning is an approach that can be used for extracting value out of data, through training of a machine learning model. By utilizing machine learning, it is possible for an enterprise to use all the data that it has to run analysis on it to find patterns and other hidden data.
Extracting value out of data can help an enterprise become more proactive. It can do so by training a prediction model which can anticipate future happenings such as errors and faults. If an enterprise is able to extract value from its existing data then it will be able to make a prediction model. Based on the predictions from the prediction model, an enterprise can act proactively to anticipated events.

29.1.6 Prototype evaluation conclusion

This section will conclude the discussion about the prototype developed.

Even though the author might not be too satisfied with the results of the prototype based on the data available, the opinion is still that the prototype and the development of it can be useful for enterprises. The reason it can still be useful for enterprises is that the prototype can be used as a first tutorial for predictive maintenance. The code for doing the predictive maintenance is included in appendix E, and it can work independent of the data that is inserted. The program is general and can be customized for each enterprise that wants to do predictive maintenance. The prototype was meant as a tool to evaluate the *framework for proactive enterprises*, but also to guide other enterprises that would like to be proactive how to start when it comes to predictive maintenance.

The advantage that Business A can have from the prototype is limited. The reason is that the number of features included in the model in the end was so low, given the amount of initial features. Since there were a lot of missing values in the dataset used for the prediction model training, most of the features were removed. The prototype might still be useful for the enterprise, but since the prototype only can handle a low amount of the features that should be included, it will be limited. This could have been evaluated with an integration of the prediction model in the architecture of Business A, but this was not done since there was a lack of time for doing so. The prototype was evaluated purely quantitative through the testing phase of machine learning, while the value for Business A is to be seen.

29.2 Framework evaluation

In this section, the author will discuss and evaluate the work done with enhancing the framework.

29.2.1 Framework capabilities

During the work with this master thesis, it became clear that several of the capabilities defined could have been merged to simplify the framework while still holding its initial power.

It can be discussed whether the capabilities of awareness and perceptivity can be merged. The same approaches are not used for these two capabilities as awareness uses approaches such as SWOT, SOAR and SCOPE analysis as well as data analysis, while perceptivity only uses data analysis. It might not be correct that these should be merged either, but they both concern the ability to understand data or situations. The awareness capability focuses more on the situation an enterprise are in, while perceptivity is focused on making meaning out of the data an enterprise gather and possess. It might be better to leave these two apart, but a more clear border might be needed to separate them.

A second couple of capabilities that is up for discussion are the intelligence and prediction capabilities. As prediction is about being able to foresee future events and intelligence are about making a smart decision based on the information an enterprise has available, these should have a reasonably clear border to separate them. On the other hand, it might be argued that the prediction capability should be a step within the intelligence capability. As of today, the prediction capability consists of only one step, which is either does an enterprise utilize prediction or they do not. The fact that the same technologies can be used within both of them makes a case for that prediction should be a subpart within the intelligence capability. It would be needed a thorough process before a merge can be done between the two capabilities.

The decisions for whether some of the capabilities should be merged will be left for future work, as it is outside the scope of this project and the time has been prioritized elsewhere in the form of identifying approaches and technologies for each of them. It can very well be argued that the order should have been reversed, but the reason these possible merging tasks are more apparent now is that of the work done during this master thesis. It could be said that the possible merge tasks would not have been as clear if work was done in another order.

29.2.2 General discussion

Several approaches and technologies are identified through the work of this master thesis. A vast amount of relevant articles have been processed, and a part of the literature review, to end up with the result gotten. More approaches and technologies are available for each of the capabilities which have not been identified, but the amount of work thoroughly identifying approaches and technologies for all capabilities turned out too extensive and time-consuming. It can be argued that some of the capabilities should have been prioritized while the work with the rest of them could have been left for future work, but the primary focus of this thesis was to enhance the framework including all aspects of it. In hindsight, it might have been wiser to focus on 3-5 of the capabilities to be able to go more in-depth on each of them. If this was done then a better description and small tutorials could have been made for every one of them. Another fact that resulted in a more narrow result of the literature review is the time that was used for developing the prototype.

By developing the prototype, evaluation of the framework was made possible. During the development of the *framework for proactive enterprises*[Ham17], it was stated that business A needed to improve the proactive aspect. The enhanced framework developed during this master thesis also contributed in the way that it suggested technologies and approaches that could be used to improve the enterprise. By developing the prototype, even though the results of the prototype was not as good as initial hopes, the effect on the enterprise is illustrated through framework and is presented in section 27.2.

The following paragraphs discusses shortly when an enterprise should use the capabilities.

For the different capabilities, there are different times of when an approach or technology should be used. Many of the capabilities within the framework are very different, and it would not make sense to introduce new approaches and technologies to them all at the same time.
When it comes to decentralization of an enterprise, a good time to do this would be when the enterprise is proliferating, or the industry that the enterprise is within is changing. By having a decentralized enterprise, it is easier to adapt to changes in a quicker manner.
Introduction of approaches and technologies within interoperability should come at a time when an enterprise is growing and introducing new systems which are meant to communicate well with the other enterprise systems. Interoperability is something that should be in the mind of an enterprise from the very beginning, and the stricter it is followed, the easier it will be to maintain when the enterprise grows. For extroversion, the tools identified should be introduced at a stage where several enterprises are starting a collaboration to finish a product they usually wouldn't be able to do by themselves.
Awareness is also a capability that enterprises should try to maintain at a high level. The tools identified for this capability should be used by an enterprise when they want to increase their awareness, either internally or externally. If an enterprise can stay situationally aware, they can detect changes, opportunities, and threats in the market they operate in a faster way.
An enterprise should invest in tools and techniques for perceptivity when they are interested in analyzing and understanding better the data they generate. An optimal time for an enterprise to introduce this aspect is if they have the data amounts required by analysis tools.
Predicting future events can be very useful for enterprises, and techniques and technologies for being able to do that should be introduced when an enterprise wants to stay ahead of the competitors and when they want to save money and time on aspects such as maintenance. Prediction is used for a lot more than maintenance, and an enterprise can take advantage of the technologies to predict the market, sales numbers and so on.
An enterprise should introduce intelligent decision support systems when they are in need of quicker decisions or systems that can aid their decision makers. The systems can improve based on the information that flows through them and by doing that they become more intelligent.
When an enterprise wants to automate their processes and optimize them, they should introduce automated processes in their product line or in their business processes to streamline their work.

How an enterprise can introduce some of the approaches and technologies can best be answered through the prototype that was developed. For the author to understand and gain knowledge on how all the different approaches and technologies can be used would have been a too broad scope. For that reason, a single aspect of an enterprise, the proactive aspect, was tested. The proactive aspect consists of the capabilities prediction, intelligence, and action. The approach that was tested was machine learning and the technology used was Amazon Sagemaker.
The description of how testing and developing of this prototype were done through the use of this technology can be found in part VI of this thesis, while the guidebook and tutorial for the specific program developed can be found in appendix E.

29.2.3 Framework evaluation conclusion

This section will conclude the discussion about the enhanced version of the framework.

As have been discussed here, there might be several of the capabilities within the framework that can change in the future. Some of the capabilities can be merged to optimize the value of them and to simplify the framework. The enhanced framework consists of approaches and technologies for all of the existing capabilities as of today, and the prototype helped illustrating the enhancement of an enterprise through the use of the framework.
There is work that still needs to be done to the framework to retrieve the maximized potential out of it, but this master thesis has enhanced the framework further than what it was at the beginning of this project.

30 Research methodology evaluation

This chapter will evaluate the research methodologies used.

30.1 Method

During the work with the thesis, several methods are used. For gathering information and prepare for the prototype developed there was used structured interviews. For researching the enhancement of the *framework for proactive enterprises*, there was used literature review and then more specifically keyword search and domain expert contact to gather relevant articles.

30.1.1 Structured interviews

The templates for the structures interviews conducted can be found in appendices A, B and C.

Structured interviews have advantages and disadvantages.
Structured interviews are well fitted when it is not possible to record the interview session, as a structured interview makes the interviewee answer on exactly what is the questions, and they do not have as much freedom. The possibility of the interviewee talking about irrelevant information is less likely with a structured interview. The interview templates specified questions which needed to be answered, and questions besides those were not discussed.
Structured interviews can in many cases help get quantifiable data, but that is in cases where there are many interviewees. As there was only Business A to interview, the focus was on getting quality for the answers, and to learn more about why the contribution could help them. A positive consequence of a structured interview is the ability to make sure the interviewee fully understands the questions. When the author had the interviews with Business A with the defined questions, it was easier for them to answer precisely and short as he had specified the questions to a certain degree. While doing the interviews, structured interviews, it is easier for the interviewer to stick to the questions at hand instead of letting the conversation flow and drift away from the vital information.

A negative consequence of a structured interview is that the interviewee might not be able to express and tell the interviewer everything they would like to tell. For the author, this was not a big concern as he thought well through what he wanted to get out of each interview before conducting them. By doing that he made sure that there was a minimal amount of information that the author might have missed. During interviews, the personal values and mindset of each interviewee can affect the answers they give. In this case, as the answers that were possible to give to each question was more based on technical aspects, the interviewees could not let personal values affect the answers to a counting degree.

To conclude, structured interviews worked very well in this case. It was possible to retrieve the information needed on a fast pace, and the information obtained was specified enough so that it was relevant for this thesis and the part of the thesis concerning the prototype development.

30.1.2 Literature review

While doing the literature review there were used two different approaches to gather and collect relevant information. Those approaches were keyword search and domain expert contact, while also setting up criteria for which articles or papers would be included.

Keyword search is an efficient way to reduce a set of articles within a field. The number of articles online these days are massive, and it can be overwhelming when a researcher tries to find information about a specific field. Where do the researcher start, how can he know which articles are more relevant as well as other questions are needed to be thought through. While using keyword search and specifying different groups of keywords that are synonyms, it is possible to end up with a much smaller set size than just searching randomly for papers. For the author, it was an excellent technique to use, since then he could reduce the number of papers to read concerning every capability of the framework. Since the framework consists of eight different capabilities, there were still a lot of articles and papers to read.

The project supervisor helped find articles in the form of domain expert contact. She was available for the author if he had a hard time finding articles and papers within some fields, and contributed with her knowledge to find papers to review. By utilizing her expertise, it was possible to find relevant information to study faster than if the author needed to find it himself.

Using these techniques still lead to a considerable amount of articles and papers, which made the author insert criteria for which would be included in the final set. Utilizing criteria reduced the relevant set a lot and made it easier to conduct the literature review.

To conclude, the different techniques used helped reduce the set of relevant articles, but there was still a lot to process. The literature review ended up contributing to enhancing the framework by being the primary source of information.

30.1.3 Prototype development

For the prototype development, *agile development* was used through the technique of *scrum.* As mentioned earlier, the author did all of the development by himself, and for that reason, scrum meetings were naturally not held traditionally. The client was not included regularly in the process either.
However, sprint planning was done and sprint review as well. The development process went fine in this project, and scrum was a successful approach. By splitting the development into smaller sprints, the task did not feel overwhelming at any time, and the author knew precisely what needed focus.

One of the lessons learned from the development period is that things took longer time than expected. This is a valuable lesson to bring along as the author has experienced the same in other relevant scenarios within computer science.
Another lesson learned is that using a method like scrum was useful, even when the author was developing alone.

31 Evaluation of project

In this chapter, the author will discuss how the research questions were answered and how the project goal was reached. Every one of the research questions will be discussed individually, including the subquestions. The project goal will be discussed in a more general manner.

31.1 Research questions

In this section, the author will answer specifically for each research question how they were answered.

31.1.1 RQ1 Which technologies exists for supporting the different capabilities within the *framework for proactive enterprises*?

To identify technologies that exist for supporting the different capabilities within the framework, a thorough literature review was done, which can be found in part V. For some of the capabilities, it was easier to find relevant technologies than for some of the others. For some of the capabilities, the author was able to find real technology products which can directly help an enterprise transforming already at an implementation level. This applies to the capabilities decentralization, perceptivity,

prediction, and intelligence. Within these capabilities, the author could find specific technologies such as team communication platforms as Slack and Microsoft teams as well as others, additionally to machine learning and complex event processing tools through Amazon Sagemaker and Oracle CEP including others.
For other capabilities, it was only possible to identify frameworks and tools which could help enterprises. These frameworks and tools were found for the capabilities interoperability and extroversion. For the awareness capability a technique, more than a technology was found through the use of sensor data to gather and analyze information.No technologies were identified for the action capability.

The author was also able to test one of the technologies found by making a prototype using Amazon Sagemaker to train a prediction model based on data from Business A for predictive maintenance. The prototype is in connection with **RQ4** in section 31.1.4.
The amount of time spent on developing the prototype for Business A limited the literature review. The development of the prototype exceeded the planned time, and for that reason, less time was spent on the literature review resulting in a narrower result for the literature review.

The results found for both **RQ1** and **RQ2** can be found in table 21.

31.1.2 RQ2 Which approaches are used for the different capabilities?

Approaches for the same capabilities were identified through literature review as well. The author was able to identify approaches for every one of the capabilities. For decentralization approaches such as spreading decision making, sharing of knowledge between the different departments and setting restrictions on what the departments could do was identified as approaches.
Concerning interoperability, techniques and approaches to make systems communicate better were introduced. By utilizing communication protocols and standardization of messages, an enterprise can make sure that systems can communicate with each other. Another approach that can be used is to take advantage of using a broker or a middle-ware which will make sure the different systems can understand each other even without following the same standards. Three different approaches were also presented in the form of integrated, unified and federated approach which all have a different degree of how to implement interoperability.
For awareness, tools such as SWOT, SCOPE, and SOAR among others were identified. By utilizing these tools, an enterprise can increase internal, external and situational awareness.
An enterprise can increase its perception of data by utilizing data analysis, mainly through historical and real-time data. Through the use of data analysis, an enterprise can find patterns in data which are not apparent to the human eye and then get a better understanding of the data they possess.
Approaches identified for the capability of prediction are machine learning and complex event processing. Both of these approaches can be used by an enterprise to try and predict future happenings and events. For an enterprise to increase their intelligence, decision support systems were introduced as an approach and then specifically intelligent decision support systems which utilize AI to make better decisions.
The approaches identified that can be used by an enterprise for the action capability are either human-based action or automated processes. An example where automated processes can be used to a high degree is within smart warehouses or production lines.
For the capability extroversion, approaches such as open innovation, co-creation and social technologies were defined.

The results found for both **RQ1** and **RQ2** can be found in table 21.

31.1.3 RQ3 Can the results found by addressing RQ1 and RQ2 enhance the *framework for proactive enterprises*?

The results that are found for **RQ1** and **RQ2** can help enhance the framework. The way that the results enhance the framework is by providing a more straightforward understanding for enterprise managers on how they can start their journey towards being proactive by being presented with the approaches and technologies that can be used within the different capabilities. The previous version of the framework mostly contributed to helping an enterprise identify which capabilities that needed improvement to advance as an enterprise, and which steps should be taken. It did not specify *how* an enterprise could do the steps specified. The enhanced version of the framework contributes to specifying, within the different capabilities, which approaches and technologies can be used to achieve the goal of an enterprise, thus enhancing the *how* of the framework.

31.1.4 RQ4 How can developing a prototype for predictive maintenance using machine learning help enhance the *framework for proactive enterprises*?

By developing a prototype for predictive maintenance using machine learning it was possible to enhance the framework by now including the prototype to it. The prototype contributes to the framework by presenting a specific example of how an enterprise can be more proactive. The framework is supposed to help enterprises understand how and where they can improve, and the prototype answers the *how* for the capabilities intelligence and prediction. Several prototypes could have been developed to enhance the framework, but in this thesis, a prototype developed with machine learning was prioritized.

A more specific example of how the prototype contributes to the framework can be presented with an imagined scenario. Imagine that an enterprise manager or similar is using the framework as he or she wants to improve the enterprise they work in. The manager might want to improve the prediction and intelligence capabilities and can get from the framework which technologies can be used. He does not know how to use them, but the prototype can guide him through the development of predictive maintenance, by utilizing the Jupyter Notebook in appendix E.

31.1.5 RQ5 How is it possible to create value from existing data to make enterprises more proactive?

An enterprise can gain several advantages by being proactive. By being proactive, an enterprise can better handle the pressure towards the quality of their products. They are more prepared for changes that can happen in the environments they work in and they will be able to take advantage of external opportunities in a better way. These opportunities and advantages can sometimes be found through the data they possess.

An enterprise can become more proactive by extracting value out of data that it has. By analyzing the data and fully understand it, it is possible to collect value from it. An approach that can be used for extracting value from data to become more proactive is through machine learning. By utilizing machine learning, an enterprise can analyze its data and find hidden patterns.
Training the processed data and making a prediction model helps an enterprise become more proactive. The prediction model can be used to anticipate future events and an enterprise can act proactively according to the anticipated events.

Part IX
Conclusion and future work

This part will be the final part of this thesis, and it will conclude the project. The part consists of two chapters, where chapter 32 will conclude the thesis by including the essential information from the project. Chapter 33 will present thoughts about future work that can be done to enhance the framework further.

32 Conclusion

This chapter will conclude the master thesis and the research done concerning the enhancement of the *framework for proactive enterprises*. The framework is meant to be used by enterprise managers or whomever it might interest to clarify and map the current situation and position of an enterprise. By using the framework, the enterprise can become aware of which steps are needed to be taken to advance as an enterprise and transform into being a proactive enterprise.

In this master thesis, a literature review was conducted to enhance the framework. Approaches and technologies within the different capabilities in the framework were researched to get a better view on what are the possibilities an enterprise has. Identifying approaches and technologies makes it possible for the framework to, more specifically, suggest how an enterprise can advance within the different capabilities. For some of the capabilities, other frameworks and tools were identified.

A prototype was developed to enhance a collaborating enterprise illustrated through the framework. In addition, it enriches the framework by adding an example of an approach and technology. A prediction model for predictive maintenance was developed through Amazon Sagemaker. A collaborating enterprise provided the data with status reports from their enterprise machines. By analyzing this data and making a prediction model based on it, an improvement to the enterprise could be seen in the framework. Due to missing data, the actual prediction model can probably not be used to its full potential by the enterprise, but the prototype still contributes to the enhancement of the framework. A description of how the prediction model was developed is included and can be used by other enterprises which want to start with predictive maintenance. The description can work as an introduction or guidebook for how they can start with predictive maintenance.

The project presents an enhanced framework in the form of adding approaches and technologies to each capability that can be used by enterprises. The contribution makes sure that it should be clearer *how* an enterprise can move forward towards being a proactive enterprise. Before this project, the framework answered the questions about *what* needed to be done by an enterprise and *why* they should improve. After this project, the framework is also able to answer the question *how* an enterprise can make specific measures to advance.

Research question 1 asked which technologies existed for each of the capabilities within the framework at hand. Through literature review, different technologies were identified which could be used within the different capabilities of the framework. It was not possible to identify technology products for all of the capabilities, and for some of them, frameworks were identified instead.

Research question 2 asked a similar question as research question 1, but it focused on which approaches were available for each capability instead of technologies. Through literature review, it was possible to identify different approaches for the different capabilities. For some of the capabilities, it was possible to identify tools that could be used, while for some it was possible to find general approaches.

Research question 3 asked whether the results found through research question 1 and 2 could contribute to enhancing the framework. This master thesis shows that it was possible to enhance the framework through the work done.

Research question 4 asked how the development of a prototype could help enhance the framework. The prototype was developed to evaluate how the framework can help an enterprise advance. Even given the weaker results of the prototype, the framework showed that with a successful introduction of such a prediction model would advance the enterprise.

Research question 5 asked how it is possible to create value from existing data to make enterprises more proactive. The thesis discusses this question and concludes that it can be done through approaches such as data analysis, machine learning and making prediction models.

The main project goal of the thesis was:

The project goal is to enhance the "framework for proactive enterprises".

done through several subgoals:

- Conduct a literature review of emerging and relevant technologies
- Enhance the *framework for proactive enterprises*
- Identify a relevant technology and methodology to develop a prototype
- Develop and evaluate the prototype

The subgoals of the project were achieved through the literature review done and the development of the prototype. The combination of these subgoals resulted in achieving the main project goal through enhancing the *framework for proactive enterprises*. The framework could have been enhanced in several different directions, where introducing approaches and technologies to accompany it was one of them. The framework still has much potential where it can be enhanced, but concerning the state, the framework was in before the master thesis and after, it can be said that the framework indeed got enhanced.

33 Future work

Future work will consist of enhancing the framework further, so it is of a better use for enterprises. As mentioned in this thesis, there are several of the capabilities that need better research and that some merging tasks might be done. For this merging tasks to be done, thorough work and analysis should be done to make sure that no information within the framework gets lost in the process. Future work can also consist of further identification of approaches and technologies that can be used. Further identification will help enhance the framework and increase the interest towards it.

With the final words of this thesis, the author would like to open the use and enhancing tasks for everyone. There is no restriction on who can use the framework, and there is no restriction on who can use the framework as a basis for their work.

References

[AAL+07] Daron Acemoglu, Philippe Aghion, Claire Lelarge, John Van Reenen, and Fabrizio Zilibotti. Technology, information, and the decentralization of the firm. *The Quarterly Journal of Economics*, 122(4):1759–1799, 2007.

[ACZ13] Denise E Agosto, Andrea J Copeland, and Lisl Zach. Testing the benefits of blended education: Using social technology to foster collaboration and knowledge sharing in face-to-face lis courses. *Journal of education for library and information science*, pages 94–107, 2013.

[A.N17] Srikanth A.N. Slack alternatives: 10 best tools for team communication, 2017. [Online; accessed February 26, 2018].

[AP17] Roshan Alwis and Srinath Perera. Machine learning techniques for predictive maintenance, 2017. [Online; accessed January 24, 2018].

[AWS18] AWS. Machine learning with amazon sagemaker, 2018. [Online; accessed March 9, 2018].

[BDLB14] Sergej Baranenko, Mihail Dudin, Nikolay Lyasnikov, and Konstantin Busygin. Use of environmental approach to innovation-oriented development of industrial enterprises. 2014.

[BEF+07] A-J Berre, Brian Elvesæter, Nicolas Figay, Claudia Guglielmina, Svein G Johnsen, Dag Karlsen, Thomas Knothe, and Sonia Lippe. The athena interoperability framework. In *Enterprise interoperability II*, pages 569–580. Springer, 2007.

[BGB] Beryl Bellman, Ken Griesi, and Mark Bergman. Agile enterprise architecture: Oxymoron or new vision.

[BK09] Alejandro Buchmann and Boris Koldehofe. Complex event processing. *IT-Information Technology Methoden und innovative Anwendungen der Informatik und Informationstechnik*, 51(5):241–242, 2009.

[BPM+16] Alexandros Bousdekis, Nikos Papageorgiou, Babis Magoutas, Dimitris Apostolou, and Gregoris Mentzas. Continuous improvement of proactive event-driven decision making through sensor-enabled feedback (sef). In *ICEIS (2)*, pages 166–173, 2016.

[Bra17] Mary Branscombe. Microsoft teams review: Serious competition for slack and hipchat, 2017. [Online; accessed February 26, 2018].

[BSVR10] Nicholas Bloom, Raffaella Sadun, and John Van Reenen. Does product market competition lead firms to decentralize? *American Economic Review*, 100(2):434–38, 2010.

[CD03] David Chen and Guy Doumeingts. European initiatives to develop interoperability of enterprise applications—basic concepts, framework and roadmap. *Annual reviews in control*, 27(2):153–162, 2003.

[CDV08] David Chen, Guy Doumeingts, and François Vernadat. Architectures for enterprise integration and interoperability: Past, present and future. *Computers in industry*, 59(7):647–659, 2008.

[Cha06] K Mani Chandy. Event-driven applications: Costs, benefits and design approaches. *Gartner Application Integration and Web Services Summit*, 2006, 2006.

[Cha16] David Chappell. Understanding machine learning, 2016. [Online; accessed December 17, 2017].

[Che06] David Chen. Enterprise interoperability framework. In *EMOI-INTEROP*, 2006.

[Chu16] Douglas Okafor Chukwuekwe. Condition monitoring for predictive maintenance:-a tool for systems prognosis within the industrial internet applications. Master's thesis, NTNU, 2016.

[CRVP01] Kartik Chandran, Sudarshan Raghunathan, Subbarayan Venkatesan, and Ravi Prakash. A feedback-based scheme for improving tcp performance in ad hoc wireless networks. *IEEE Personal communications*, 8(1):34–39, 2001.

[CZW+16] Qiming Chen, Bin Zhang, Ren Wu, Andy Therber, Meichun Hsu, Hans Zeller, and David Birdsall. Map-reduce and parallel processing in databases, February 23 2016. US Patent 9,268,815.

[DCV05] Nicolas Daclin, David Chen, and Bruno Vallespir. Design principles and pattern for decisional interoperability. In *proceedings of IFIP*, volume 5, 2005.

[Dha13] Vasant Dhar. Data science and prediction. *Communications of the ACM*, 56(12):64–73, 2013.

[Dou17] Eric Douglas. Using values to decentralize decision-making, 2017. [Online; accessed February 19, 2018].

[DQR12] Jens Dittrich and Jorge-Arnulfo Quiané-Ruiz. Efficient big data processing in hadoop mapreduce. *Proceedings of the VLDB Endowment*, 5(12):2014–2015, 2012.

[DR13] Alina Dulipovici and Daniel Robey. Strategic alignment and misalignment of knowledge management systems: A social representation perspective. *Journal of Management Information Systems*, 29(4):103–126, 2013.

[Duf18] Jill Duffy. Slack, 2018. [Online; accessed February 21, 2018].

[eCCO12] Carlos A Bana e Costa, María Carmen Carnero, and Mónica Duarte Oliveira. A multi-criteria model for auditing a predictive maintenance programme. *European Journal of Operational Research*, 217(2):381–393, 2012.

[FAB+11] Matthias Farwick, Berthold Agreiter, Ruth Breu, Steffen Ryll, Karsten Voges, and Inge Hanschke. Automation processes for enterprise architecture management. In *Enterprise Distributed Object Computing Conference Workshops (EDOCW), 2011 15th IEEE International*, pages 340–349. IEEE, 2011.

[Fii18] Fiix. What is predictive maintenance?, 2018. [Online; accessed January 29, 2018].

[fin12] Future internet enterprise systems - research roadmap 2025, 2012. [Online; accessed November 25, 2017].

[fM15] AzureML Team for Microsoft. Predictive maintenance: Step 1 of 3, data preparation and feature engineering, 2015. [Online; accessed January 24, 2018].

[Fre18] Roslyn Frenz. Decentralized information management strategies, 2018. [Online; accessed February 18, 2018].

[FVC15] Kelwin Fernandes, Pedro Vinagre, and Paulo Cortez. A proactive intelligent decision support system for predicting the popularity of online news. In *Portuguese Conference on Artificial Intelligence*, pages 535–546. Springer, 2015.

[G+98] C4ISR Architecture Working Group et al. Levels of information systems interoperability (lisi). *US DoD*, 1998.

[GBM00] Stephanie Guerlain, Donald E Brown, and Christina Mastrangelo. Intelligent decision support systems. In *Systems, man, and cybernetics, 2000 ieee international conference on*, volume 3, pages 1934–1938. IEEE, 2000.

[Gis11] Gis. Interoperability, 2011. [Online; accessed February 28, 2018].

[Goe16] Vera Goebel. Complex event processing (cep), 2016. [Online; accessed May 16, 2018].

[Gooa] Google Dictionary. Action. [Online; accessed November 24, 2017].

[Goob] Google Dictionary. Awareness. [Online; accessed November 24, 2017].

[Gooc] Google Dictionary. Intelligence. [Online; accessed November 24, 2017].

[Good] Google Dictionary. Interoperability. [Online; accessed November 24, 2017].

[Gooe] Google Dictionary. Perceptive. [Online; accessed November 24, 2017].

[Goof] Google Dictionary. Predict. [Online; accessed November 24, 2017].

[Gui] Getting Started Guide. Oracle® complex event processing.

[GVCK18] Philip Gautier, Cyrus Vahid, Saswata Chakravarty, and Zohar Karnin. Train faster, more flexible models with amazon sagemaker linear learner, 2018. [Online; accessed April 25, 2017].

[Had13] Hadoop. Mapreduce tutorial, 2013. [Online; accessed May 15, 2018].

[Ham17] Eirik Hamnvik. A framework for helping enterprises position themselves towards future enterprises. *Norwegian University of Techonology and Science*, 2017.

[Has11] HM Hashemian. Wireless sensors for predictive maintenance of rotating equipment in research reactors. *Annals of Nuclear Energy*, 38(2-3):665–680, 2011.

[HB11] Hashem M Hashemian and Wendell C Bean. State-of-the-art predictive maintenance techniques. *IEEE Transactions on Instrumentation and measurement*, 60(10):3480–3492, 2011.

[HW97] Terry Hill and Roy Westbrook. Swot analysis: it's time for a product recall. *Long range planning*, 30(1):46–52, 1997.

[IBM] IBM. Predictive maintenance. [Online; accessed January 30, 2018].

[Inc18] TIBCO Software Inc. Streaming analytics for accelerating action, 2018. [Online; accessed May 23, 2018].

[JKL07] Håvard D Jørgensen, Dag Karlsen, and Frank M Lillehagen. Product based interoperability-approaches and requirements. In *CCE*, pages 33–44, 2007.

[Kau16] David Kauchak. Neural networks. 2016.

[KG09] Kevin A Kaiser and Nagi Z Gebraeel. Predictive maintenance management using sensor-based degradation models. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 39(4):840–849, 2009.

[KM17] Eamonn Keogh and Abdullah Mueen. Curse of dimensionality. In *Encyclopedia of Machine Learning and Data Mining*, pages 314–315. Springer, 2017.

[Kol16] Swetha Kolalapudi. How to think about machine learning algorithms, 2016. [Online; accessed December 14, 2017].

[KP12] Anders Kofod-Petersen. How to do a structured literature review in computer science. *Ver. 0.1. October*, 1, 2012.

[Kur16] Jerry Kurata. Understanding machine learning with python, 2016. [Online; accessed December 14, 2017].

[Kur17] Jerry Kurata. Tensorflow: Getting started, 2017. [Online; accessed December 14, 2017].

[LAK+15] Fenareti Lampathaki, Heroon Polytechniou Athens, Panagiotis Kokkinakos, Christopher Tucci, Iosif Alvertis, John Psarras, Sotiris Koussouris, and Gianluigi Viscusi. Futureenterprise, a roadmap for sensing enterprise. 2015.

[Lis12] Pierre Lison. An introduction to machine learning. 2012.

[LP17] Dongjin Lee and Rong Pan. Predictive maintenance of complex system with multi-level reliability structure. *International Journal of Production Research*, 55(16):4785–4801, 2017.

[LPH+14] Hongfei Li, Dhaivat Parikh, Qing He, Buyue Qian, Zhiguo Li, Dongping Fang, and Arun Hampapur. Improving rail network velocity: A machine learning approach to predictive maintenance. *Transportation Research Part C: Emerging Technologies*, 45:17–26, 2014.

[LRSS03] Mikael Lindvall, Ioana Rus, and Sachin Suman Sinha. Software systems support for knowledge management. *Journal of knowledge management*, 7(5):137–150, 2003.

[Luc02] David Luckham. *The power of events*, volume 204. Addison-Wesley Reading, 2002.

[LW06] Christine Legner and Kristin Wende. Towards an excellence framework for business interoperability. *BLED 2006 Proceedings*, page 29, 2006.

[Lă17] Tudor Lăpușan. Hadoop mapreduce deep diving and tuning, 2017. [Online; accessed May 15, 2018].

[Mal04] Thomas V. Malone. Making the decision to decentralize, 2004. [Online; accessed February 19, 2018].

[Maz16] Elizabeth Mazenko. Predictive maintenance: How it works and why you need it, 2016. [Online; accessed January 30, 2018].

[MBY+16] Xiangrui Meng, Joseph Bradley, Burak Yavuz, Evan Sparks, Shivaram Venkataraman, Davies Liu, Jeremy Freeman, DB Tsai, Manish Amde, Sean Owen, et al. Mllib: Machine learning in apache spark. *The Journal of Machine Learning Research*, 17(1):1235–1241, 2016.

[Mic18a] Microsoft. Meet microsoft teams, the hub for teamwork in office 365, 2018. [Online; accessed February 26, 2018].

[Mic18b] Microsoft. Professional online meetings built for business, 2018. [Online; accessed February 26, 2018].

[MPCAF15] Miguel A Martínez-Prieto, Carlos E Cuesta, Mario Arias, and Javier D Fernández. The solid architecture for real-time management of big semantic data. *Future Generation Computer Systems*, 47:62–79, 2015.

[MS16] Mihnea Alexandru Moisescu and Ioan Stefan Sacala. Towards the development of interoperable sensing systems for the future enterprise. *Journal of Intelligent Manufacturing*, 27(1):33–54, 2016.

[MSB+14a] Babis Magoutas, Nenad Stojanovic, Alexandros Bousdekis, Dimitris Apostolou, Gregoris Mentzas, and Ljiljana Stojanovic. Anticipation-driven architecture for proactive enterprise decision making. In *CAiSE (Forum/Doctoral Consortium)*, pages 121–128, 2014.

[MSB+14b] Babis Magoutas, Nenad Stojanovic, Alexandros Bousdekis, Dimitris Apostolou, Gregoris Mentzas, and Ljiljana Stojanovic. Anticipation-driven architecture for proactive enterprise decision making. In *CAiSE (Forum/Doctoral Consortium)*, pages 121–128, 2014.

[MWBR11] Paul Mathiesen, Jason Watson, Wasana Bandara, and Michael Rosemann. Applying social technology to business process lifecycle management. In *International Conference on Business Process Management*, pages 231–241. Springer, 2011.

[Nam15] Dmitry Namiot. On big data stream processing. *International Journal of Open Information Technologies*, 3(8), 2015.

[NRZ14] Ovidiu Noran, David Romero, and Milan Zdravkovic. The sensing enterprise: Towards the next generation dynamic virtual organisations. In *Working Conference on Virtual Enterprises*, pages 209–216. Springer, 2014.

[Pet14] Sobah Abbas Petersen. Interoperability lecture. 2014.

[PJH16] CM Park, HY Jang, and JH Han. What characteristics make an enterprise more proactive toward innovation? case of adopting rfid at kumho tire in korea. In *Management of Innovation and Technology (ICMIT), 2016 IEEE International Conference on*, pages 309–312. IEEE, 2016.

[Pow00] Daniel J Power. Web-based and model-driven decision support systems: concepts and issues. *AMCIS 2000 Proceedings*, page 387, 2000.

[Pow08] Daniel J Power. Understanding data-driven decision support systems. *Information Systems Management*, 25(2):149–154, 2008.

[PS07] Daniel J Power and Ramesh Sharda. Model-driven decision support systems: Concepts and research directions. *Decision Support Systems*, 43(3):1044–1061, 2007.

[Pug16] Jean Francois Puget. The machine learning workflow, 2016. [Online; accessed January 29, 2018].

[PvdKP16] Sobah Abbas Petersen, Rimmert van der Kooij, and Primoz Puhar. Connecting business processes and sensor data in proactive manufacturing enterprises. In *Working Conference on Virtual Enterprises*, pages 101–109. Springer, 2016.

[RCL14] Reza Rezaei, Thiam Kian Chiew, and Sai Peck Lee. A review on e-business interoperability frameworks. *Journal of Systems and Software*, 93:199–216, 2014.

[Riv16] Matthew Lamare Rachel Rivera. *Comparative Analysis of Outsourcing Railcar Maintenance*. PhD thesis, Worcester Polytechnic Institute, 2016.

[RL02] Ioana Rus and Mikael Lindvall. Knowledge management in software engineering. *IEEE software*, 19(3):26, 2002.

[Rob10] D Robins. Complex event processing. In *Second International Workshop on Education Technology and Computer Science. Wuhan*, pages 1–10. Citeseer, 2010.

[SAP] SAP. Move from reactive to predictive maintenance and service – with the internet of things. [Online; accessed January 30, 2018].

[Sch97] Ken Schwaber. Scrum development process. In *Business object design and implementation*, pages 117–134. Springer, 1997.

[Sci17] Elite Data Science. Master machine learning, 2017. [Online; accessed January 29, 2018].

[SCK03] Jacqueline Stavros, David Cooperrider, and D Lynn Kelley. Strategic inquiry appreciative intent: inspiration to soar, a new framework for strategic planning. *AI Practitioner. November*, pages 10–17, 2003.

[Scr18] Scrum.org. What is scrum?, 2018. [Online; accessed April 23, 2018].

[Ser18a] Amazon Web Services. Amazon emr, 2018. [Online; accessed May 15, 2018].

[Ser18b] Amazon Web Services. Evaluating ml models, 2018. [Online; accessed April 9, 2018].

[Ser18c] Amazon Web Services. What is amazon sagemaker?, 2018. [Online; accessed May 15, 2018].

[s.F02] Lalitha s.Fernando. Theoretical approach to decentralization: An introduction. *Journal of Management Science*, 1:117–141, 2002.

[Sim18] Alex Simons. Azure ad b2b collaboration for hybrid organizations, 2018. [Online; accessed June 7, 2018].

[SKRC10] Konstantin Shvachko, Hairong Kuang, Sanjay Radia, and Robert Chansler. The hadoop distributed file system. In *Mass storage systems and technologies (MSST), 2010 IEEE 26th symposium on*, pages 1–10. Ieee, 2010.

[SN08] Rainer Schmidt and Selmin Nurcan. Bpm and social software. In *International Conference on Business Process Management*, pages 649–658. Springer, 2008.

[SS17] Veda C Storey and Il-Yeol Song. Big data technologies and management: What conceptual modeling can do. *Data & Knowledge Engineering*, 108:50–67, 2017.

[Sta17] Eric Starker. Azure ad b2b collaboration in microsoft teams, 2017. [Online; accessed June 7, 2018].

[Tay17] Ian Taylor. Microsoft skype for business review – a bite sized intro, 2017. [Online; accessed February 26, 2018].

[tea18a] Office 365 team. Use office 365 sharepoint online as a business-to-business (b2b) extranet solution, 2018. [Online; accessed June 7, 2018].

[tea18b] Slack team. Slack, 2018. [Online; accessed February 21, 2018].

[Tec12] TechGig. What is complex event processing?, 2012. [Online; accessed May 16, 2018].

[The] The PACE Project. Decentralization. [Online; accessed November 24, 2017].

[Too16] Lean Manufactoring Tools. Predictive maintenance, 2016. [Online; accessed January 30, 2018].

[Tri16] Anir Tripathi. The difference between predictive maintenance and preventive maintenance, 2016. [Online; accessed January 29, 2018].

[Ven17] David Venturi. Every single machine learning course on the internet, ranked by your reviews, 2017. [Online; accessed January 29, 2018].

[VHP13] Adriaan Van Horenbeek and Liliane Pintelon. A dynamic predictive maintenance policy for complex multi-component systems. *Reliability Engineering & System Safety*, 120:39–50, 2013.

[VM14] Jinesh Varia and Sajee Mathew. Overview of amazon web services. *Amazon Web Services*, 2014.

[Wal15] Michael Walker. Machine learning workflow, 2015. [Online; accessed January 29, 2018].

[Wik18] Wikiwand. Enterprise interoperability framework, 2018. [Online; accessed May 10, 2018].

[Win18] Beth Winston. Alternatives to swot analysis, 2018. [Online; accessed March 5, 2018].

[WM96] George Wyner and Thomas Malone. Cowboys or commanders: Does information technology lead to decentralization? *ICIS 1996 Proceedings*, page 4, 1996.

[WMC+16] Georg Weichhart, Arturo Molina, David Chen, Lawrence E Whitman, and François Vernadat. Challenges and current developments for sensing, smart and sustainable enterprise systems. *Computers in Industry*, 79:34–46, 2016.

[Woo18a] Tessa Wooley. Azure active directory b2b collaboration for hybrid organizations, 2018. [Online; accessed June 7, 2018].

[Woo18b] Tessa Wooley. What is azure ad b2b collaboration?, 2018. [Online; accessed June 7, 2018].

[Wun17] Dusty Wunderlich. 3 ways to decentralize management and boost productivity, 2017. [Online; accessed February 21, 2018].

[Yan08] Jeffrey Yan. Social technology as a new medium in the classroom. *New England Journal of Higher Education*, 22(4), 2008.

[YLL+09] Ching-man Au Yeung, Ilaria Liccardi, Kanghao Lu, Oshani Seneviratne, and Tim Berners-Lee. Decentralization: The future of online social networking. In *W3C Workshop on the Future of Social Networking Position Papers*, volume 2, pages 2–7, 2009.

[YTLT01] RCM Yam, PW Tse, L Li, and P Tu. Intelligent predictive decision support system for condition-based maintenance. *The International Journal of Advanced Manufacturing Technology*, 17(5):383–391, 2001.

[Zha14] Ying Zhang. Benefits of automated production lines, 2014. [Online; accessed May 16, 2018].

[ZWLS15] Zhigao Zheng, Ping Wang, Jing Liu, and Shengli Sun. Real-time big data processing framework: challenges and solutions. *Applied Mathematics & Information Sciences*, 9(6):3169, 2015.

[ZXW+16] Matei Zaharia, Reynold S Xin, Patrick Wendell, Tathagata Das, Michael Armbrust, Ankur Dave, Xiangrui Meng, Josh Rosen, Shivaram Venkataraman, Michael J Franklin, et al. Apache spark: a unified engine for big data processing. *Communications of the ACM*, 59(11):56–65, 2016.

Part X
Appendices

This part includes the appendices that come with the master thesis. Appendix A presents the evaluation form that was used during the evaluation of Business A. Appendix B presents an interview template that was used during a structured interview to plan further towards the development of the prototype, while appendix C shows the template for the interview done when identifying why the prototype could be necessary for the enterprise.
Appendix D further explains the Java program developed to clean and process the enterprise data, while appendix E includes the Jupyter Notebook that was developed to do further data cleaning and the actual training and testing of the prediction model.

A Enterprise evaluation form template

Enterprise Evaluation Form	
Enterprise:	
Contact Person:	
Date:	
Description	**Value**
Decentralization	
The enterprise is centralized, with a top-down approach	0
Some or all of the enterprise systems have decentralized intelligence	4
The enterprise consists of self-organizing teams with a flat hierarchy	8
The enterprise consist of self-organizing teams where workers are in need of continuous learning	12
Interoperability	
No interoperability	0
Electronic exchange of analog and digital data	3
Enterprise systems are able to collaborate on a functional level	6
Systems are able to do coordinated service exchange	9
Multiple enterprise systems are capable of acting together	12
Awareness	
No awareness	0
Systems are aware of conditions and status of different parts within themselves	4
Internal enterprise awareness can include organizational structure, learning capability and technical infrastructure	8
Environmental awareness can include position in market, government support and new business opportunities	12
Perceptivity	
The enterprise does not perceive data	0
The enterprise analyses and interprets historical and knowledge domain data	6
Real-time analysis of data provided by heterogeneous sources	12
Prediction	
No prediction	0
The enterprise is anticipating the future based on real-time and historical data	12
Intelligence	
No intelligence	0
Behavior patterns are discovered and stored to influence future decisions	4
Intelligent systems makes decisions based on business goals and situational awareness	8
Decision making system in continuous improvement through sensor-enabled feedback	12
Action	
Actions are taken reactively	0
Actions are taken proactively by humans	6
Actions are taken automatically by systems	12
Extroversion	
The enterprise is based on closed innovation and knowledge ownership	0
Blurry boundaries - Introduces communication with external sources	4
Sharing information to embrace innovations and collaboration processes	8
A high degree of integration between partners to secure alignment of individual plans to achieve a joint goal	12

Table 33: Enterprise evaluation form template

B Enterprise prototype development planning sheet

Enterprise Prototype Planning
Enterprise:
Contact Person:
Date:
What type of data could I receive from your company?
How will we be able to transfer the data?
For how long of a period do you have data from that can be given to me?
How many machines have delivered status reports for the data I'm about to receive?
Are there some legal documents that needs to be signed for me to use the data?

Table 34: Enterprise Prototype Planning

C Predictive maintenance interview template

Predictive Maintenance - Interview
Enterprise:
Contact Person:
Date:
Which consequences surface for your company if one of your machines goes out of service?
Which consequences surface for your customers if one of your machines goes out of service?
Which type of maintenance is done for your machines as of today? **How often, in general, does your machines need maintenance?**
How much data is produced per day per operating machine? **How often does your machines send status updates?**
How much does it cost to schedule and perform maintenance on one of your machines as of today?
How far ahead is maintenance required to be scheduled?
How much of the budget is spent on maintenance costs?

Table 35: Predictive maintenance interview template

D Data handling Java program

To prepare and handle the data there was developed a Java program which would analyze and connect the data. The data received came in the form of several different status report types, and all of these were put into separate folders. Within each folder, the status reports were organized into folders depending on dates. The data was received as zipped files, and first all of the files needed to be unzipped before going into the program. The status reports are formatted as CSV files and needed to be read into the program.

In the program, classes were made corresponding to the different status report types. A class was made for an RVM, and the RVM class had a list of status reports. The program took a folder URL in and went through all files and under folders within it, reading every line in every file. Depending on the type of status report that was being analyzed different methods were invoked to handle the status report correctly. The status reports are connected by looking at which machine they came from and at what time they were sent. By doing this, it was possible to have more information of what happened at a specific machine at a specific time.

When all of the folders and files are gone through and analyzed the program takes the list of RVMs that it contains and generates a new text file formatted as CSV. The CSV file is now containing all the vital information found in the different files analyzed for each machine at the same time and date.

When the program is finished running through the data, the author has a CSV file which is ready to be put directly into the machine learning algorithms at AWS.

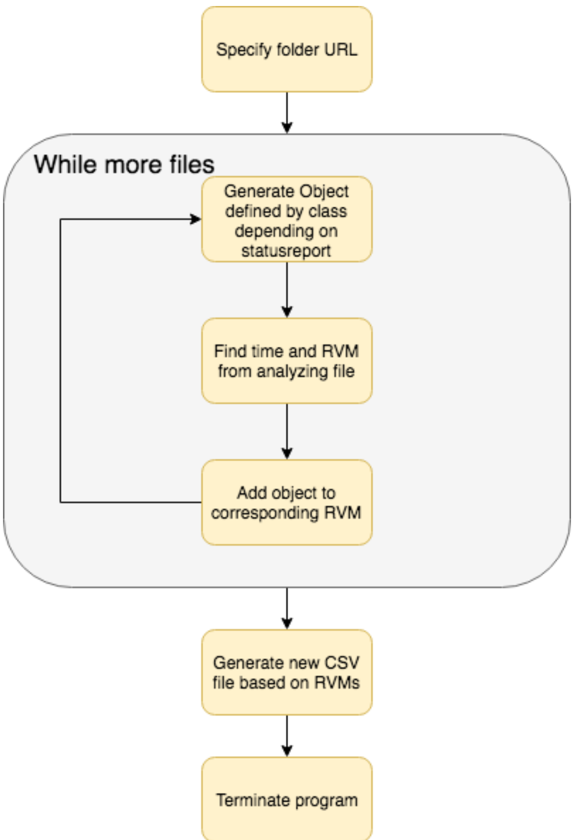


Figure 23: Java program flow chart

E AWS Jupyter Notebook

Predictive maintenance prototype

This is a prototype for predictive maintenance and its purpose is to help enterprises who wants to start their journey towards being proactive. The prototype is developed using data from an enterprise which contains enterprise secret data and is for that reason sencored.

This document will consist of several steps to make the data usable and be of meaning to a machine learning algorithm. The document will also show how to train a machine learning model using a built-in option for an algorithm from Amazon Web Services through Amazon Sagemaker.

The problem at hand is a supervised binary classification problem where the prediction model will predict if an enteprise machine is likely to fail in the near future.

Prerequisites

An enterprise who wants to start with predictive maintenance needs a lot of data to use for training. In the data used, the real outcome will also need to be included. In the case of predictive maintenance it is important whether the machine was running at the point or not.

During development of this prototype, our data was uploaded in a CSV-format to a Amazon S3 bucket. It is recommended to store data there for easy access from the sagemaker.

Setup

First we connect to our S3 bucket to retrieve the data.

In [1]:

```
from sagemaker import get_execution_role

role = get_execution_role()
bucket='[redacted]' #replace with your own bucket name
```

Fetching data

We do imports and fetch our data from the s3 bucket

In [2]:

```
import boto3
import io
import matplotlib.pyplot as plt
import numpy as np
import os
import pandas as pd

import sagemaker
import sagemaker.amazon.common as smac
from sagemaker import get_execution_role
from sagemaker.predictor import csv_serializer, json_deserializer

prefix = "sagemaker"
s3_train_key = '{}/train/business_b_data'.format(prefix)
s3_train_path = os.path.join('s3://', bucket, s3_train_key)
s3 = boto3.client('s3')
obj = s3.get_object(Bucket=bucket, Key='csv_final_anonymous2.csv') #key is the c
sv file to import
df = pd.read_csv(io.BytesIO(obj['Body'].read()))
```

Print out the head of the dataframe to see the import went fine

In [3]:

```
df.head() #Print out head of dataframe to see that the outcome is as expected
```

Out[3]:

	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	...	
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	I
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	I
2	[redacted]	2.0	[redacted]	-1.0	1.0	337417.0	NaN	1.0	2.0	43.962	...	I
3	[redacted]	2.0	[redacted]	24.0	1.0	2431698.0	NaN	1.0	1.0	NaN	...	I
4	[redacted]	2.0	[redacted]	71.0	1.0	160155.0	NaN	1.0	7.0	0.666	...	I

5 rows × 57 columns

We see that we have 57 columns to begin with.

Data preparation

This stage consists of several steps that are taken to handle missing values and information that is not useful for a machine learning algorithm. The data needs to be prepared before the machine learning algorithm can accept it.

Feature selection

First we fix our columns which contains our features.

Remove columns which are not useful

The columns removed were just used during data preparation from original data and will not be of use here.

In [4]:

```
df = df.drop(columns=['v1', 'v3', 'v7'])
df.head() #Print out head of dataframe to see that the outcome is as expected
```

Out[4]:

	v2	v4	v5	v6	v8	v9	v10	v11	v12	v13	...	v48	v49	v50
0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
1	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
2	2.0	-1.0	1.0	337417.0	1.0	2.0	43.962	0.004	25.0	6.0	...	NaN	NaN	NaN
3	2.0	24.0	1.0	2431698.0	1.0	1.0	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
4	2.0	71.0	1.0	160155.0	1.0	7.0	0.666	0.000	7.0	2.0	...	NaN	NaN	NaN

5 rows × 54 columns

We see that our set of columns have been reduced to 54.

Remove all rows that have a 100% missing values

There is no point to keep the rows that contains only missing values, and for that reason we remove them from our dataframe.

In [5]:

```
df = df.dropna(how='all')
df.head() #Print out head of dataframe to see that the outcome is as expected
```

Out[5]:

	v2	v4	v5	v6	v8	v9	v10	v11	v12	v13	...	v48	v49	v50	v51
2	2.0	-1.0	1.0	337417.0	1.0	2.0	43.962	0.004	25.0	6.0	...	NaN	NaN	NaN	0.0
3	2.0	24.0	1.0	2431698.0	1.0	1.0	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN
4	2.0	71.0	1.0	160155.0	1.0	7.0	0.666	0.000	7.0	2.0	...	NaN	NaN	NaN	NaN
5	2.0	-1.0	2.0	219309.0	1.0	7.0	2.266	0.005	NaN	NaN	...	NaN	NaN	NaN	0.0
6	1.0	-1.0	1.0	379476.0	3.0	0.0	25.000	0.000	4.0	2.0	...	NaN	NaN	NaN	0.0

5 rows × 54 columns

Remove columns that have more than 50% NaN

Columns that have a big amount of missing data will not be included in the training and will therefore be removed

In [6]:

```
df = df.dropna(thresh=len(df)/2, axis=1)
df.head() #Print out head of dataframe to see that the outcome is as expected
```

Out[6]:

	v2	v4	v5	v6	v8	v9	v10	v11	v12	v13	v16	v57
2	2.0	-1.0	1.0	337417.0	1.0	2.0	43.962	0.004	25.0	6.0	0.50	0.0
3	2.0	24.0	1.0	2431698.0	1.0	1.0	NaN	NaN	NaN	NaN	NaN	0.0
4	2.0	71.0	1.0	160155.0	1.0	7.0	0.666	0.000	7.0	2.0	0.09	0.0
5	2.0	-1.0	2.0	219309.0	1.0	7.0	2.266	0.005	NaN	NaN	0.17	0.0
6	1.0	-1.0	1.0	379476.0	3.0	0.0	25.000	0.000	4.0	2.0	0.39	1.0

We see that our set of columns have been reduced to 12.

Amount of remaining columns, excluding the target column

We store the value because it is required as a hyperparameter for our algorithm at a later point We remove the count of the target column as it is not counted as a feature

In [7]:

```
feature_dim_value = len(df.columns) - 1
feature_dim_value #Print out the value
```

Out[7]:

11

Handling missing values

Replace all NaN values with 0's

Machine learning algorithms needs int values and missing values are replaced with 0

In [8]:

```
df = df.fillna(0)
df.head() #Print out head of dataframe to see that the outcome is as expected
len(df) #Check how many rows we are left with
```

Out[8]:

124772

Data convertion

Convert all values to float32, as it is required by the machine learning algorithm

In [9]:

```
df = df.apply(pd.to_numeric)
df.head() #Print out head of dataframe to see that the outcome is as expected
```

Out[9]:

	v2	v4	v5	v6	v8	v9	v10	v11	v12	v13	v16	v57
2	2.0	-1.0	1.0	337417.0	1.0	2.0	43.962	0.004	25.0	6.0	0.50	0.0
3	2.0	24.0	1.0	2431698.0	1.0	1.0	0.000	0.000	0.0	0.0	0.00	0.0
4	2.0	71.0	1.0	160155.0	1.0	7.0	0.666	0.000	7.0	2.0	0.09	0.0
5	2.0	-1.0	2.0	219309.0	1.0	7.0	2.266	0.005	0.0	0.0	0.17	0.0
6	1.0	-1.0	1.0	379476.0	3.0	0.0	25.000	0.000	4.0	2.0	0.39	1.0

Check that all types are of desired data type

In [10]:

```
df.dtypes
```

Out[10]:

```
v2      float64
v4      float64
v5      float64
v6      float64
v8      float64
v9      float64
v10     float64
v11     float64
v12     float64
v13     float64
v16     float64
v57     float64
dtype: object
```

Check data

How many running instances and how many not running instances

In [11]:

```
dff = df.groupby('v57').size() #v57 is our target column
dff.head() #Print out head of dataframe to see that the outcome is as expected
```

Out[11]:

```
v57
0.0    102801
1.0     21971
dtype: int64
```

Calculate the percentage of running and not running instances

In [12]:

```
df.v57.value_counts(normalize=True)
```

Out[12]:

```
0.0    0.823911
1.0    0.176089
Name: v57, dtype: float64
```

We can see here that the data is very unbalanced. This problem can be fixed by the built-in algorithm from Amazon which we will see later.

Prepare data for training

In [13]:

```
# Read the data, shuffle, and split into train and test sets, separating the lab
els (last column) from the features
raw_data = df.as_matrix()
np.random.seed(0)
np.random.shuffle(raw_data)
train_size = int(raw_data.shape[0] * 0.7)
train_features  = raw_data[:train_size, :-1]
train_labels = raw_data[:train_size, -1]
test_features = raw_data[train_size:, :-1]
test_labels = raw_data[train_size:, -1]

# Convert the processed training data to protobuf and write to S3 for linear lea
rner
vectors = np.array([t.tolist() for t in train_features]).astype('float32')
labels = np.array([t.tolist() for t in train_labels]).astype('float32')
buf = io.BytesIO()
smac.write_numpy_to_dense_tensor(buf, vectors, labels)
buf.seek(0)
boto3.resource('s3').Bucket(bucket).Object(s3_train_key).upload_fileobj(buf)
```

In [14]:

```
vectors #We check that our data was split successfully and that our vectors look
 fine
```

Out[14]:

```
array([[  2.  ,  -1.  ,   1.  , ...,   0.  ,   0.  ,   0.2 ],
       [  1.  ,  -1.  ,   1.  , ...,   0.  ,   0.  ,   0.25],
       [  2.  , 100.  ,   1.  , ...,  18.  ,  14.  ,   0.19],
       ...,
       [  2.  ,  -1.  ,   1.  , ...,   0.  ,   0.  ,   0.28],
       [  2.  ,  91.  ,   1.  , ...,   3.  ,   1.  ,   0.16],
       [  2.  , 100.  ,   3.  , ...,  10.  ,   3.  ,   0.28]],
      dtype=float32)
```

In [15]:

```
labels #We check that our data was split successfully and that the labels vector
 looks fine
```

Out[15]:

```
array([0., 0., 1., ..., 1., 0., 0.], dtype=float32)
```

Define functions to be used during training, evaluation and clean-up

Function for training a prediction model based our training data and a set of defined hyperparameters

The function will also create an endpoint which will be used for evaluation of the prediction model

In [16]:

```
def predictor_from_hyperparams(s3_train_data, hyperparams, output_path):
    """
    Create an Estimator from the given hyperparams, fit to training data, and re
turn a deployed predictor
    """

    instance_type_chosen = 'ml.m4.xlarge'

    # specify algorithm containers and instantiate an Estimator with given hyper
params
    containers = {
        'us-west-2': '174872318107.dkr.ecr.us-west-2.amazonaws.com/linear-learne
r:latest',
        'us-east-1': '382416733822.dkr.ecr.us-east-1.amazonaws.com/linear-learne
r:latest',
        'us-east-2': '404615174143.dkr.ecr.us-east-2.amazonaws.com/linear-learne
r:latest',
        'eu-west-1': '438346466558.dkr.ecr.eu-west-1.amazonaws.com/linear-learne
r:latest'}
    linear = sagemaker.estimator.Estimator(containers[boto3.Session().region_nam
e],
        role,
        train_instance_count=1,
        train_instance_type= instance_type_chosen,
        output_path=output_path,
        sagemaker_session=sagemaker.Session())
    linear.set_hyperparameters(**hyperparams)
    # train model
    linear.fit({'train': s3_train_data})
    # deploy a predictor
    linear_predictor = linear.deploy(initial_instance_count=1, instance_type= in
stance_type_chosen)
    linear_predictor.content_type = 'text/csv'
    linear_predictor.serializer = csv_serializer
    linear_predictor.deserializer = json_deserializer
    return linear_predictor
```

Function for evaluating a prediction model comparing the test features to the actual test labels

In [17]:

```
def evaluate(linear_predictor, test_features, test_labels, model_name, verbose=F
alse):
    """
    Evaluate a model on a test set given the prediction endpoint.  Return binary
 classification metrics.
    """
    # split the test data set into 100 batches and evaluate using prediction end
point
    prediction_batches = [linear_predictor.predict(batch)['predictions'] for bat
ch in np.array_split(test_features, 100)]
    # parse raw predictions json to exctract predicted label
    test_preds = np.concatenate([np.array([x['predicted_label'] for x in batch])
 for batch in prediction_batches])

    # calculate true positives, false positives, true negatives, false negatives
    tp = np.logical_and(test_labels, test_preds).sum()
    fp = np.logical_and(1-test_labels, test_preds).sum()
    tn = np.logical_and(1-test_labels, 1-test_preds).sum()
    fn = np.logical_and(test_labels, 1-test_preds).sum()

    # calculate binary classification metrics
    recall = tp / (tp + fn)
    precision = tp / (tp + fp)
    accuracy = (tp + tn) / (tp + fp + tn + fn)
    f1 = 2 * precision * recall / (precision + recall)

    if verbose:
        print(pd.crosstab(test_labels, test_preds, rownames=['actuals'], colname
s=['predictions']))
        print("\n{:<11} {:.3f}".format('Recall:', recall))
        print("{:<11} {:.3f}".format('Precision:', precision))
        print("{:<11} {:.3f}".format('Accuracy:', accuracy))
        print("{:<11} {:.3f}".format('F1:', f1))

    return {'TP': tp, 'FP': fp, 'FN': fn, 'TN': tn, 'Precision': precision, 'Rec
all': recall, 'Accuracy': accuracy,
             'F1': f1, 'Model': model_name}
```

Function for deleting an endpoint made during training

This will delete an endpoint which makes takes the prediction model go out of use.

In [18]:

```
def delete_endpoint(predictor):
        try:
            boto3.client('sagemaker').delete_endpoint(EndpointName=predictor.end
point)
            print('Deleted {}'.format(predictor.endpoint))
        except:
            print('Already deleted: {}'.format(predictor.endpoint))
```

Train prediction model

The training will be done in two different rounds which will create two prediction models and two endpoints for testing.

Logistic

The first round we train a binary classifier with default settings. We pass in the feature_dim_value variable which we defined earlier specifying how many features we have in our data. We include that the predictor_type should be binary classification and that the training should run through the data 40 times (default is 10).

In [ ]:

```
# Training a binary classifier with default settings: logistic regression
defaults_hyperparams = {
    'feature_dim': feature_dim_value,
    'predictor_type': 'binary_classifier',
    'epochs': 40
}
defaults_output_path = 's3://{}/{}/defaults/output'.format(bucket, prefix)
#Call the training function to do the actual training with the specified hyperpa
rameters
defaults_predictor = predictor_from_hyperparams(s3_train_path, defaults_hyperpar
ams, defaults_output_path)
```

Logistic with class weights

The secound round we train a binary classifier with class weights and automated threshold tuning. The class weights takes care of the problem we had with unbalanced data.

In [ ]:

```
# Training a binary classifier with class weights and automated threshold tuning
class_weights_hyperparams = {
    'feature_dim': feature_dim_value,
    'predictor_type': 'binary_classifier',
    'binary_classifier_model_selection_criteria': 'precision_at_target_recall',
    'target_recall': 0.9,
    'positive_example_weight_mult': 'balanced',
    'epochs': 40
}
class_weights_output_path = 's3://{}/{}/class_weights/output'.format(bucket, pre
fix)
#Call the training function to do the actual training with the specified hyperpa
rameters
class_weights_predictor = predictor_from_hyperparams(s3_train_path, class_weight
s_hyperparams, class_weights_output_path)
```

Testing model

We run evaluation on all of our defined models and display a table containing the values for recall, precision, accuracy and F1.

In [21]:

```
# Evaluate the trained models
predictors = {'Logistic': defaults_predictor,
              'Logistic with class weights': class_weights_predictor}
metrics = {key: evaluate(predictor, test_features, test_labels, key, False) for
key, predictor in predictors.items()}
pd.set_option('display.float_format', lambda x: '%.3f' % x)
display(pd.DataFrame(list(metrics.values())).loc[:, ['Model', 'Recall', 'Precisi
on', 'Accuracy', 'F1']])
```

	Model	Recall	Precision	Accuracy	F1
0	Logistic	1.000	0.982	0.997	0.991
1	Logistic with class weights	0.951	1.000	0.991	0.975

Clean-up

Delete all of the endpoints we have created

In [22]:

```
for predictor in [defaults_predictor, class_weights_predictor]:
    delete_endpoint(predictor)
```

```
Deleted linear-learner-2018-04-25-11-21-51-732
Deleted linear-learner-2018-04-25-11-35-43-732
```