

Lappeenranta University of Technology  
LUT School of Business and Management  
Industrial Engineering and Management  
Business Analytics

Mikko Ritala

## **Detection and data-driven root cause analysis of paper machine drive anomalies**

Author: Mikko Ritala

Examiners: Professor Pasi Luukka

Post Doctoral Researcher Jan Stoklasa

Supervisors: Director of Analytics & Applications Development Arttu-Matti Matinlauri

# ABSTRACT

Lappeenranta University of Technology  
LUT School of Business and Management  
Industrial Engineering and Management  
Business Analytics

Mikko Ritala

## **Detection and data-driven root cause analysis of paper machine drive anomalies**

Master's thesis

2019

81 pages, 34 figures, 13 tables, and 10 appendices

Examiners: Professor Pasi Luukka, Post Doctoral Researcher Jan Stoklasa

Keywords: Industrial Internet, predictive maintenance, machine learning, artificial intelligence, anomaly detection, root cause analysis, data preprocessing, model interpretation

The Industrial Internet has increased interest in the collection and utilization of data. The latter has become easier due to increased computing power and the development of analytical methods. The goal of this thesis is to develop methods for detecting anomalies and identifying their root causes. Machine learning (ML) is used to create regression models for the electricity consumption of paper machine (PM) drives. ML enables a way of creating models describing the behavior of equipment. In the empirical part of the thesis, ML models are compared to the physics-based model. Creating a physics-based model requires considerable knowledge of the process. Considering all the process features that affect electricity consumption is a time-consuming task. The ML model, on the other hand, learns the effects of process features from historical data.

Anomalies are identified by comparing model output and measured process value. Time periods when the difference between model output and measured process value is significant are considered anomalous. During an anomalous time period, there may have been an undesired change in the process that could lead to equipment damage. Process features that can explain anomalies are sought from the data using the Pearson correlation. Knowing what caused the anomalies can help to prevent machine failures.

An application is created during the thesis that is used in the case study in which machine failures are studied. A model is created to find anomalous periods, and the application is used to identify the root causes of anomalies. The application can explain anomalies, but root causes for machine failures cannot be identified. In future research, other methods for root cause identification besides correlation could be studied.

# TIIVISTELMÄ

Lappeenrannan teknillinen yliopisto  
LUT School of Business and Management  
Tuontantotalous  
Business Analytics

Mikko Ritala

## **Paperikoneen linjakäyttöjen anomalioiden tunnistus ja juurisyyanalyysi**

Diplomityö

2019

81 sivua, 34 kuvaa, 13 taulukkoa ja 10 liitettä

Tarkastajat: Professori Pasi Luukka, Tutkijatohtori Jan Stoklasa

Hakusanat: Teollinen Internet, ennakoiva kunnossapito, koneoppiminen, tekoäly, poikkeavuuksien havaitseminen, juurisyyanalyysi, datan esikäsittely, mallin tulkinta

Teollinen Internet on lisännyt kiinnostusta datan keräämiseen ja hyödyntämiseen. Laskentatehon ja analyttisten menetelmien kehitys on helpottanut datan hyödyntämistä. Tavoitteena tässä työssä on kehittää menetelmiä poikkeavuustilanteiden tunnistamiseen ja niiden juurisyiden selvittämiseen. Työssä rakennetaan paperikoneen käyttöjen sähkön kulutusta mallintavia regressiomalleja koneoppimisen avulla. Koneoppimista käyttämällä voidaan tehdä erilaisten laitteiden toimintaa kuvaavia malleja. Työn empiirisessä osuudessa koneoppimismalleja verrataan fysiikkaan perustuvaan malliin. Fysiikkaan perustuvan mallin tekeminen vaatii paljon prosessiosaamista. Kaikkien sähkönkulutukseen vaikuttavien tekijöiden huomioon ottaminen vie paljon aikaa. Koneoppimismalli oppii prosessinmuuttujien väliset vuorovaikutukset historiadatasta.

Poikkeavuudet tunnistetaan vertailemalla mallin tulosta mitattuun prosessiarvoon. Ajanjaksoja pidetään poikkeavina, kun mallin tuloksen ja mitatun prosessiarvon erotus on merkittävä. Poikkeavan ajanjakson syynä voi olla epäedullinen muutos prosessissa. Muutos prosessissa voi pahimmassa tapauksessa johtaa laitteiden hajoamiseen. Selittäviä tekijöitä poikkeavuustilanteille etsitään datasta Pearson korrelaation avulla. Tieto siitä, mikä aiheutti poikkeaman voi auttaa vikatilanteiden ehkäisemisessä.

Työn aikana rakennettiin sovellus, jota hyödynnetään vikatilanteiden tutkimisessa. Ensin rakennetaan malli poikkeavuuksien löytämiseksi, minkä jälkeen juurisyytä etsitään sovelluksen avulla. Työkalun avulla voidaan selittää poikkeavuustilanteita, mutta juurisyytä vikaantumisille ei löytynyt. Tulevaisuudessa muitakin menetelmiä kuin korrelaatiota voidaan tutkia juurisyiden etsimisessä.

# ACKNOWLEDGMENTS

First and foremost, I would like to express my greatest gratitude to Valmet for giving me this opportunity. Special recognition goes to my supervisor Arttu-Matti Matinlauri. He has done a wonderful job of creating a unique working environment within the DevOps-team. The support from the whole team and other Valmetees helped me to get through the challenges that I faced during the thesis. Thanks to Jari Kääriäinen, who always helped with papermaking-related issues. I would also like to thank Miska Valkonen, who had time to discuss ML-related topics.

I am very glad that I have reached the end of my studies at the Lappeenranta Lahti University of Technology. At the same time, I feel sad, because this chapter of my life is coming to an end. There was a point in my studies when I had a hard time find things that interested me and offered the right challenges. Without LUT and its business analytics master's program, I would not be where I am now. I, therefore, wish to thank the whole University and all my classmates for inspiring me and making my time during those years unforgettable. My thesis supervisor, Pasi Luukka, has also earned a special acknowledgment for his valuable feedback and guidance throughout the thesis. Finally, I would like to thank my family for encouraging and supporting me from day one.

# TABLE OF CONTENTS

1	Introduction.....	1
1.1	Background .....	1
1.2	Objective and scope .....	2
1.3	Execution of research .....	3
1.4	Structure of the thesis.....	3
2	Electric drives .....	5
3	Valmet Industrial Internet .....	8
4	Methodology .....	11
4.1	Data preprocessing.....	13
4.1.1	Data normalization.....	13
4.1.2	Dealing with missing values.....	14
4.1.3	Dealing with noise .....	16
4.1.4	Methods for feature selection.....	17
4.2	Machine learning models.....	23
4.2.1	Linear regression.....	23
4.2.2	Multilayer perceptron .....	24
4.2.3	Bagging .....	26
4.2.4	Boosting .....	27
4.3	Model interpretation .....	28
4.4	Root cause analysis.....	32
5	Proof of concept.....	34
5.1	Data collection .....	34
5.2	Data processing.....	35
5.3	Feature selection .....	39
5.4	Model selection.....	40

5.5	Hyperparameter tuning .....	43
5.6	Model interpretation with SHAP .....	45
5.7	Data-driven RCA .....	46
6	Application .....	50
6.1	AWS .....	50
6.2	Snowflake data storage .....	50
6.3	Tableau dashboard.....	51
6.4	Case study .....	52
7	Conclusions .....	58
8	Summary .....	60
	References .....	62
	Appendix I – SQL query .....	66
	Appendix II – Train set for physics-based model.....	67
	Appendix III – Grid search results .....	68
	Appendix IV – Cross-correlation results .....	70
	Appendix V – Cross-correlation results with time shifts .....	73
	Appendix VI – Case study: Hyperparameter grid .....	74
	Appendix VII – Case study: Grid search results .....	75
	Appendix VIII – Case study: Summary plot .....	76
	Appendix IX – Case study: Dependence plot .....	77
	Appendix X – Case study: Correlating tags of study periods .....	78

## List of symbols and abbreviations

ANFIS – Adaptive network-based fuzzy inference system  
ANN – Artificial neural network  
API – Application programming interface  
AWS – Amazon Web Services  
CBM – Condition-based monitoring  
CM – Condition monitoring  
CFS - Correlation-based feature selection  
CI/CD - Continuous integration and continuous deployment  
DAE – Deep auto-encoder  
DL – Deep learning  
EM – Expectation-maximization  
ETL – Extract, transfer, load  
IDE – Integrated development environment  
II – Industrial Internet  
IoT – Internet of Things  
KNN – K-nearest neighbor  
MAR – Missing at random  
MAJ – Majority vote  
MCAR – Missing completely at random  
MCMC – Markov Chain Monte Carlo  
MCS – Multiple classifier systems  
MICE – Multivariate imputation by chained equations  
MNAR – Missing not at random  
ML – Machine learning  
MLPR – Multilayer perceptron regressor  
MV – Missing value  
LOCF – Last observation carried forward  
PM – Paper machine  
PoC – Proof of concept  
UI – User interface  
RBAC – Role-based access control  
RCA – Root cause analysis  
ReLU – Rectified Linear Unit  
SHAP – SHapley Additive exPlanations  
SF – Snowflake  
SME – Subject matter expert  
SQL – Structured query language  
VII – Valmet Industrial Internet

# Table of Figures

Figure 2-1 Dryer group 3.....	6
Figure 2-2 Wire section.....	7
Figure 3-1 The most common services used in the VII platform.....	8
Figure 3-2 Valmet Industrial Internet platform architecture.....	9
Figure 4-1 Measurement with anomalous observations.....	11
Figure 4-2 Tabular data with MVs.....	14
Figure 4-3 Safe, borderline, and noisy observations.....	16
Figure 4-4 Three features f1 (relevant), f2 (redundant) and f3 (irrelevant).....	18
Figure 4-5 RReliefF algorithm.....	20
Figure 4-6 Pseudocode for RFE.....	22
Figure 4-7 Three-layer neural network.....	24
Figure 4-8 Single neuron.....	25
Figure 4-9 ANN training process.....	26
Figure 4-10 Adaboost algorithm.....	27
Figure 4-11 Force plot (Feature contribution in a single prediction).....	30
Figure 4-12 SHAP values of features.....	30
Figure 4-13 Dependence plot (SHAP values of a single feature).....	31
Figure 4-14 SHAP summary plot.....	32
Figure 5-1 Missing values in function of time.....	36
Figure 5-2 Missing values in function of time after elimination.....	37
Figure 5-3 Comparison of imputation methods.....	38
Figure 5-4 Test set of physics-based model.....	42
Figure 5-5 Test set of gradient boosting.....	42
Figure 5-6 Test set of linear regression.....	43
Figure 5-7 Tuned gradient boosting model during the testing period.....	44
Figure 5-8 SHAP summary plot.....	45
Figure 5-9 SHAP dependence plot.....	46
Figure 5-10 Anomalous period found with the gradient boosting method.....	47
Figure 5-11 Total power of dryer group 3 (a) and correlating tags (b) during anomaly.....	49
Figure 6-1 Tableau dashboard.....	51
Figure 6-2 Training and testing periods.....	53
Figure 6-3 Test period results of the wire section model.....	54
Figure 6-4 Study periods.....	55
Figure 6-5 Study period 3 results.....	56

## Table of Tables

Table 2-1 Electricity consumption of a modern PM. ....	5
Table 5-1 Properties of the dataset. ....	35
Table 5-2 Properties of the dataset after elimination. ....	37
Table 5-3 R-squared and MSE of the test and train set with different imputation methods..	38
Table 5-4 Train and test sets, and feature selection set. ....	39
Table 5-5 R-squared and MSE of the test and train set with different subsets.....	40
Table 5-6 Train and test sets. ....	40
Table 5-7 Performance of different models. ....	41
Table 5-8 Hyperparameter grid. ....	44
Table 5-9 Eleven most cross-correlating tags. ....	48
Table 6-1 Failures at the wire section. ....	53
Table 6-2 Maintenance action at the PM.....	54
Table 6-3 Study periods.....	55

# 1 Introduction

The term Industrial Internet (II) was invented by General Electric. Basically, the term affords an industrial perspective of the Internet of Things (IoT), which is the wider concept. The IoT encompasses four different IoT strategies: enterprise; commercial; consumer; and industrial. The IoT refers to an interaction between devices that are connected to the Internet. Data produced by these devices can be used to gain insight and improve the performance of the devices. (Alasdair 2016, 1-4)

Companies are investing increasingly in II applications every year. II applications aim to increase competitive advantage and sustainability. Condition-based maintenance (CBM) is an aspect of II that has considerable potential for development in the coming years. The goal of CBM is to identify immature equipment failures and avoid unnecessary maintenance. Maintenance decisions in CBM are done based on information collected by condition monitoring (CM). (Kumar et al. 2018)

The amount of data is continuously increasing, and utilizing this asset has raised the need for big data analytics. Working with big data is a little more challenging, because of its three main features: large volume; vast variety; and high velocity. These features make data processing crucial when doing data analysis. Big data analytics enable the obtaining of insight from massive amounts of data, which was previously impossible. ML algorithms can learn the optimal behavior of the machine from the history data gained by CM. (Lei et al. 2018; Wolfgang et al. 2017)

CM is commonly utilized in industrial processes. Due to the vast amounts of data and measurements, it is often hard to locate where the problem is when equipment is behaving abnormally. To keep production running, the time for a thorough inspection of parts is minimal. There is thus no opportunity to conduct a thorough root cause analysis (RCA). Abnormal behavior is therefore often disregarded. Maintenance is performed between fixed intervals, or when something fails. Data-driven approaches to RCA may help to avoid accidents without compromising productivity.

## 1.1 Background

Valmet launched its Industrial Internet services in 2017. The Valmet Industrial Internet (VII) framework is under heavy development. Developers are continually creating new applications. Many of the applications aim to improve performance, the quality of the end

product, cost-effectiveness, and productivity. Anomaly detection combined with data-driven RCA responds to many of these targets. Anomaly detection is not a new subject at Valmet, but it has not been greatly utilized, because it takes time to build physics-based models. The challenges arising from data are very customer-specific. ML algorithms are easier to apply with varied data. Anomaly detection with data-driven RCA is faster to implement for new customers than current solutions. It could increase the number of prevented failures, without requiring too much additional work from PM operators. (Valmet internal 2019)

Valmet's industrial history goes back more than 220 years. Valmet now operates throughout the world in process technologies, automation, and services for the pulp, paper, and energy industries. Significant investments in research and development have led Valmet to its current position. In 2018, Valmet spent 66 million euros on R&D. In that year, net sales were 3.3 billion euros, and it had more than 12,000 employees. The Industrial Internet and digitalization have been listed as Valmet's major growth accelerators. The Industrial Internet is Valmet's way of achieving one of its Must-Wins, which is customer excellence. A customer-oriented II approach guarantees that there will be a demand for II products and services. (Valmet 2018)

At Valmet, models are usually physics-based. There are many physics-based models for different parts of a PM. One use case for a physics-based model is to compare model output to measured values. If the model is sufficiently accurate, the deviations of measured values from model output may indicate a degradation of parts. Physics-based models are useful, but the potential of ML algorithms remains undiscovered. A lot of data is collected from PMs, but most of it remains unused. (Valmet internal 2019)

The reader of the thesis is expected to know the basics of data analytics and databases and the principles of statistical computing. Prior knowledge of the papermaking process is not required, but it may be helpful in interpreting the results.

## 1.2 Objective and scope

The main goal of the thesis is to explore ML model-based anomaly detection methods and identify causes of anomalies. An application is built for internal use and Valmet's customers, based on the best methods found in the empirical part of the thesis.

The main goal is achieved by completing the following objectives:

- Using data preprocessing to remove any undesired properties in data

- Utilizing feature selection methods to find an optimal set of features which help to explain the electricity consumption of drives
- Building a model for anomaly detection that can be interpreted
- Utilizing the Pearson correlation in the identification of the root causes of anomalies

The scope of this thesis consists of a complete data analysis process, from data gathering to application deployment. The focus on research is data preprocessing, ML, model interpretation, and RCA. Some limitations arise from the data security of Valmet and its customers, as well as the scarcity of literature available for similar approaches.

### 1.3 Execution of research

The project starts by becoming familiar with the existing methods found in the literature. Valmet's physics-based models work as a benchmark and as an information source for the functions of a PM. Additionally, meetings with subject matter experts (SME) were arranged to obtain a comprehensive insight into PM drives. Information from Valmet's internal sources and methods found from the literature are presented in the theoretical framework. Some of the information from internal sources is classified and cannot be presented in this thesis.

Methods introduced in the literature review are applied in the empirical part of the thesis. The VII framework is used to acquire data. SMEs help to explain the results acquired by the methods used in the thesis. Finally, the project culminates in the case study. An application is created for SMEs for studying anomalies. The application is built based on the theoretical and empirical frameworks.

### 1.4 Structure of the thesis

The thesis is divided into seven chapters. The chapters are presented in the natural order to ensure that the theoretical framework provides a basis for the empirical part of the thesis.

Theoretical framework:

- Electric drives. A short summary of drives, and the factors that influence the electricity consumption of drives in different sections at a PM. The sections where models are created are also presented.
- Valmet Industrial Internet. How data is stored, transferred, and utilized in analytics and applications at Valmet.
- Methodology. Review of methods that are used in the empirical part of the thesis for data preprocessing, ML, model interpretation, and RCA.

Empirical part:

- Proof of concept. Methods introduced in the theoretical framework are applied and compared.
- Application. How best approaches are compiled in the application and used in the case study.

The structure of the thesis follows the project's chronological order. The first chapters support an understanding of the following ones. Empirical framework conclusions are then drawn, based on the results. The last chapter is a summary that briefly describes the sections of the thesis.

## 2 Electric drives

Drives are electric motors that create rotational power to rotate the large cylinders (or rolls) of a PM. Nowadays, drives are powered by induction motors. Induction motors are the simplest, cheapest, and most reliable electric motor. These features make it the most commonly used electrical motor for producing rotary power. (Karjalainen 1999)

Electricity consumption per produced tonne of a modern PM by sections is presented in Table 2-1. Measurements are taken by Valmet from the SC PM. The total electricity used by drives is 100 kWh/t, which is 29 percent of the total electricity consumption. Generally, the speed and stretching of fabrics affect the electricity consumption of drives in every section. Paper type and basis weight determine how quickly the PM can be driven. At higher speeds, the time for dewatering decreases, so under pressure of vacuum units, it has to be increased. This creates additional friction between fabrics and vacuum units. Viscosity also increases the electricity consumption of drives when driving speed increases. (Valmet internal 2019; Karjalainen 1999)

*Table 2-1 Electricity consumption of a modern PM.*

<b>SECTION</b>	<b>ELECTRICITY CONSUMPTION</b>
<b>DRIVES AT WIRE</b>	31 kWh/t
<b>DRIVES AT PRESS</b>	48 kWh/t
<b>DRIVES AT DRYER</b>	19 kWh/t
<b>REEL</b>	2 kWh/t
<b>VACUUM SYSTEM</b>	67 kWh/t
<b>SHORT CIRCULATION</b>	79 kWh/t
<b>AIR CONDITIONING</b>	34 kWh/t
<b>POST-PROCESSING</b>	64 kWh/t
<b>TOTAL</b>	344 kWh/t

Table 2-1 shows the electricity consumption of the drives in different sections. The wire section consumes almost a third of the electricity of all the drives in the PM. The share of electricity consumption at the wire can often be higher. Most of the friction at the wire section comes from the vacuum units, which remove water from the paper web. (Karjalainen 1999)

The press section uses almost half the electricity used by the drives, but it may not always be this high. Dewatering, nip pressure, and internal energy losses from bending compensated rolls create most of the electricity consumption in the press section. (Karjalainen 1999)

During a normal run, the electricity consumption of drives in the drying section is not especially significant. Most of the power is required during the acceleration, due to the large moment of inertia at the dryer section. During a normal run, most of the power is used to overcome the friction between rolls and scrapers. Condensed water in the dryer cylinders can also significantly increase electricity consumption. (Karjalainen 1999)

Power at the reel is not significant compared with other sections. Power required at the reel comes from the required tightness of a paper web when reeling it to a roll. (Karjalainen 1999)

In the empirical part of the thesis, models are created for drives at one dryer group in a dryer section and for the wire section. Side views of the dryer group and wire section are presented in Figures 2-1 and 2-2 respectively. The dryer section consists of numerous groups of cylinders referred to as “dryer groups”. Dryer groups are heated with steam to increase the dry content of the paper web. Every dryer group is controlled as a unit, and the common power output is more important than the power output of a single drive within a dryer group. The PM operator can change the division of produced power between drives as he/she sees fit. Division change does not necessarily change the total power output. (Valmet internal 2019)

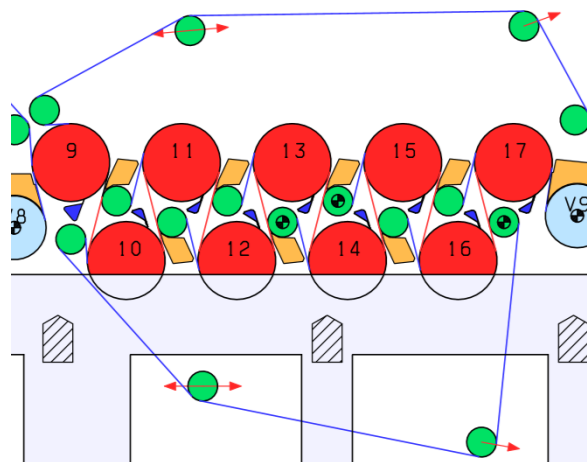


Figure 2-1 Dryer group 3 (Valmet internal 2019).

Dryer group 3, presented in Figure 2-1, has 2 fabrics (top and bottom), 15 guide rolls (green), 9 heated rolls (red, 9-17), 1 vacuum roll (blue, V9), and 4 drives, 3 in the guide rolls and 1 in the vacuum roll. The electricity consumption of the drives is not determined only within the group. The power created by adjacent groups also has an impact, which must be considered when creating a model. The following dryer groups pull the paper web, which may decrease the electricity consumption of dryer group 3. The dryer groups prior to dryer group 3 create drag, which may increase the electricity consumption of the group. (Valmet internal 2019)

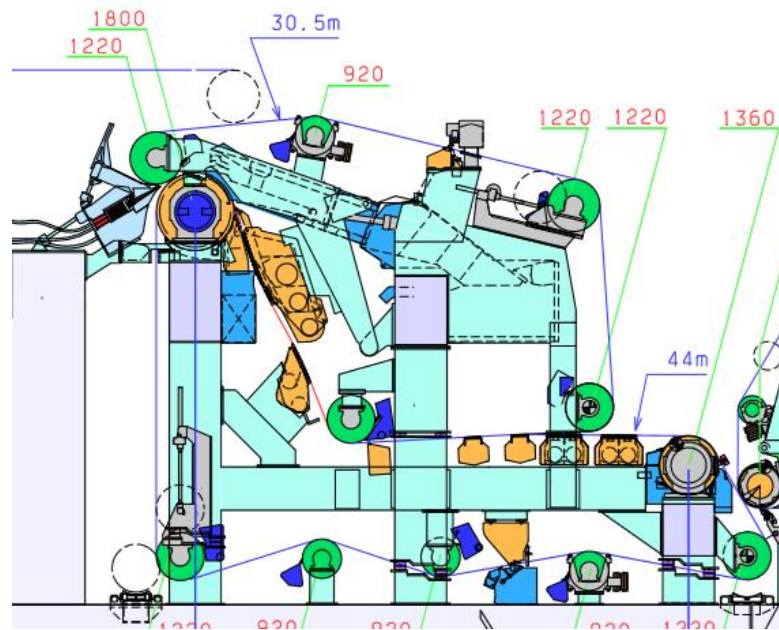


Figure 2-2 Wire section (Valmet internal 2019).

The wire section has only two drives, one for top and one for bottom fabric. The green objects are rolls, and the yellow objects are vacuum units. The stock suspension comes from a headbox to a wire section. The main purpose of a wire section is to remove water from the paper web. A wire section must also ensure that the desired structural properties of the paper are met. (Valmet internal 2019)

### 3 Valmet Industrial Internet

This chapter consists of a short summary of the VII platform and the different services used to create value from data. The chapter's objective is to introduce the reader to the services and concepts without going into unnecessary detail. The main purpose of building the VII platform is to centralize the storage and processing of the data in one place, where it can be standardized – in other words, to help the daily lives of everyone working with data. The services used in the platform are Amazon web services (AWS), Snowflake (SF), Matillion, Tableau, Python, Bitbucket, and Jenkins. Figure 3-1 shows the most commonly used services of the VII. (Valmet internal 2019)



*Figure 3-1 The most commonly used services in the VII platform.*

- AWS provides various services, such as cloud storage (S3) and serverless cloud computing (Lambda). Lambda functions can be scheduled to run on user-specified intervals. They can also be run whenever they are invoked by an event – for example, when new data is uploaded to S3.
- SF is a scalable cloud Structured Query Language (SQL) data warehouse, and it is used as a central data storage at Valmet.
- Matillion is a tool to extract, transfer, and load (ETL) data. Matillion is used to standardize customer data.
- Tableau is a dashboarding tool for creating dashboards/user interfaces (UI) for applications. Tableau can access data from cloud data storage. It is possible to do

simple calculations within Tableau, but more advanced analytics must be performed with other tools, such as Python.

- Python is a widely known programming language and the language most used by data scientists (Hayes 2019). The major advantages of Python are its simple syntax and comprehensive libraries.
- Bitbucket works as a code version control repository in which Python and other code are saved. Users can clone the source code, make changes, and push changes back to Bitbucket. The changes made are tracked in the repository, and it is always possible to go back to an earlier version.
- Jenkins is a continuous integration and continuous deployment (CI/CD) tool. Jenkins listens to a repository, such as Bitbucket, and when changes are made to the source code, Jenkins deploys changes to production. Bug fixes and new features are thus implemented to existing applications, without any additional effort.
- Other advanced analysis tools used are Alteryx and R.

Figure 3-2 shows the high-level representation of the Valmet Industrial Internet.

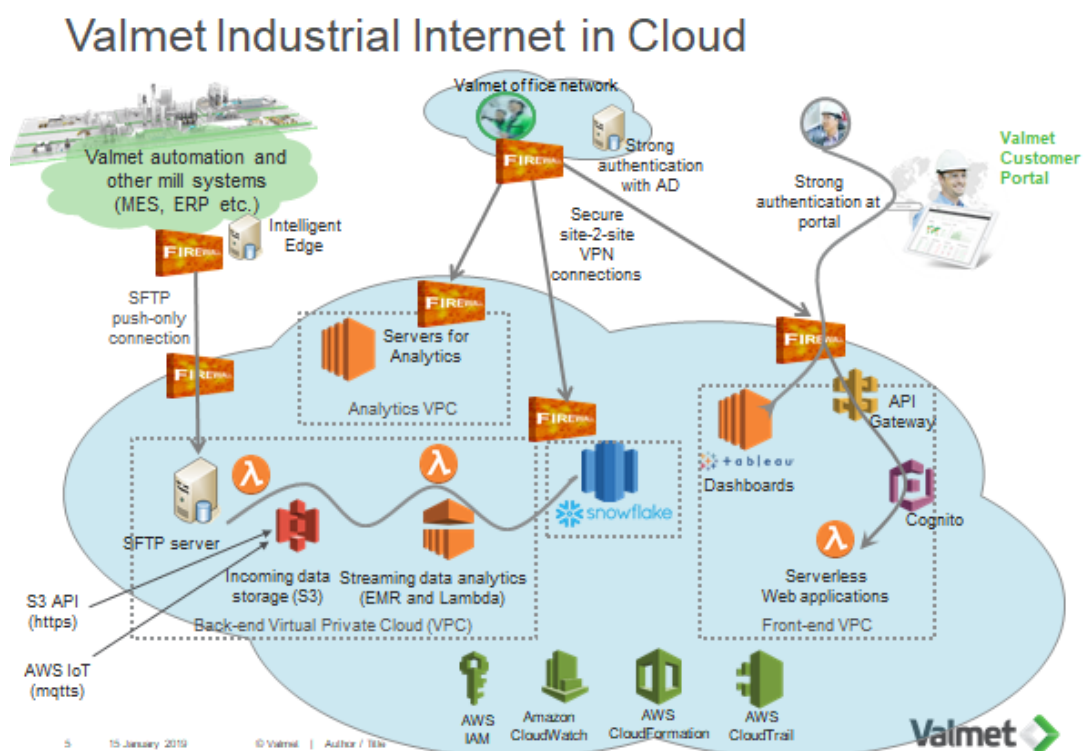


Figure 3-2 Valmet Industrial Internet platform architecture (Valmet internal 2019).

The data pipeline is the process by which data is gathered, processed, and made available for applications. Data is first gathered by sensors at the customers' machines and saved to

their local data storage, from which data is uploaded to S3. Uploading is done with R scripts, which are scheduled to upload data between fixed intervals. Now data is in S3 as batch files, where it is still quite difficult to use for analytics. Services like Matillion and Lambda are utilized to transfer and format data to SF. The data in SF is in conventional form. Simple analyses can be conducted in SF UI with SQL, or data in SF can be accessed in Tableau. SF can also be accessed with an application programming interface (API) provided by SF. SF API enables SF usage with Python or other programming languages. (Valmet internal 2019)

Lambdas are also used when creating online applications that require analysis which is impossible in Tableau or SF. For example, a VII application that compares measured values to a model output to discover if the process is working as normal can operate as follows:

1. Lambda function is triggered whenever new data is uploaded to SF
2. Lambda function downloads data from SF then calculates the model output and compares it to the measured value
3. Lambda function uploads results to SF
4. Results are visualized in Tableau dashboard

The Tableau dashboard is visible in Valmet's customer portal for authorized Valmet employees and its customers. When many customers are using the same application, access to certain data is limited by role-based access control (RBAC). (Valmet internal 2019)

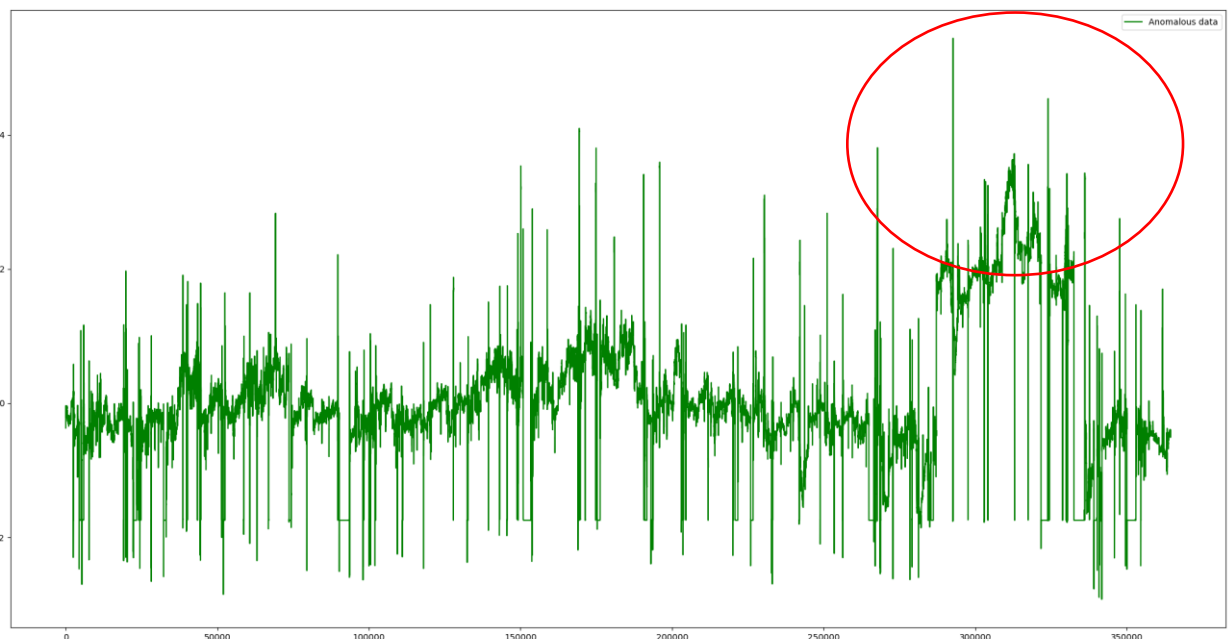
The VII platform enables fast and secure development work. Many new ideas can be brought alive quickly without using many resources. The VII platform saves time and money while creating considerable value for its users and Valmet's customers. (Valmet internal 2019)

## 4 Methodology

This chapter consists of a literature review of existing research in the field of anomaly detection, model interpretation, and the methods utilized in data-driven RCA. The aim is to cover some of the methods from the entire process, starting with data preprocessing, continuing to model building and interpretation, and finally identifying the root causes of anomalies. There are different approaches to anomaly detection, such as distance-based and clustering-based approaches. In this thesis, the focus is on model-based approaches.

Anomaly detection is not a new research area; indeed, it has been studied for more than a hundred years. The development has been rapid in recent decades, due to increases in computational power and advances in data mining. Anomalies are present in numerous disciplines. Anomalies are unexpected deviations from the norm. To detect anomalies, one must know the normal behavior of the data. Anomalies can be detected by separating anomalous observations from normal ones. This may seem an obvious classification task, but it is anything but. Anomalies also often differ from each other. This makes it difficult for classification algorithms to distinguish anomalous from non-anomalous observations. Furthermore, the number of anomalous compared to non-anomalous observations is fractional. (Mehrotra et al. 2017, 1-6)

In Figure 4-1, the anomalous behavior of the data can be seen in the marked area (red circle). The measurement receives considerably higher values than in the normal stage.



*Figure 4-1 Measurement with anomalous observations.*

Mehrotra et al. (2017, 21-22) emphasize that there are three cases to consider when assessing anomaly detection algorithms.

1. Correct Detection: Detected anomalies correspond to actual anomalies in the process.
2. False Positives: When the process is normal, even unexpected observations appear in the data. This may be due to noise.
3. False Negatives: The process deviates from the normal stage, but it is not recognized as an anomaly, because the data signal is not sufficiently significant compared to noise.

The observations in the marked area of Figure 4-1 can be classified as an anomaly. It may also be the case that the process is working as it should, but such behavior was simply unobserved in the history data. The data in Figure 4-1 exhibits low and high spikes, considered as outliers. These outliers are often caused by noise and are seen as anomalies even when the process is behaving normally. The third case may be present when noise filtering is too aggressive, and anomalies are disregarded as noise.

Underlying processes can often be described by models. The process's features have functional relationships. Since process features affect each other, it is possible to approximate one feature with a function of the other features. Anomaly detection with models can be done with two different approaches. In the first approach, the focus is on the parameters in the model and an assessment of how one model parameter affects the model output. The second approach is to compare measured data points with model outputs. The difference between the model output and a measured value is referred to as the "anomaly score". (Mehrotra et al. 2017, 57-58)

The second approach is successfully utilized in the study of Zhao et al. (2018). Their goal was the early fault detection of wind turbine components. They used a deep auto-encoder (DAE) network to model various parameters from the wind turbine. Anomalies were identified by the residuals of these parameters. They were able to detect failures more than 14 hours before actual failure. Since they modeled various variables, they were also able to locate faulty components from the residuals.

In this thesis, anomalies are studied using the second approach. The anomaly score used in the empirical part considers only moments when the measured value is higher than the model output. During those moments, there may be additional friction in the process. The anomaly score is set to zero when the model output is lower than the measured value.

## 4.1 Data preprocessing

Data preprocessing has a huge impact on prediction models since the quality of data is usually not perfect. For example, data might have missing values, outliers, noise, redundant features, or dimensionality of data is too high to be utilized effectively. This chapter is a review of basic data preprocessing methods to make data useful for ML algorithms.

### 4.1.1 Data normalization

Data normalization is usually a mandatory step before ML techniques can be applied. This is because many ML techniques are based on distances. Features with a larger distance between minimum and maximum values have more weight in prediction. To make features equal, they must be normalized. The common normalization method in the literature is min-max normalization. In min-max normalization, all features are normalized between fixed intervals. The interval used to scale the data is usually  $[0,1]$  or  $[-1,1]$ . (Gracia et al. 2015, 46-47)

Observation  $v$  of feature  $A$  is min-max normalized between range  $[a, b]$  as follows:

$$v' = \frac{v - \min(A)}{\max(A) - \min(A)} (b - a) + a, \quad (1)$$

where  $\min(A)$  and  $\max(A)$  are the minimum and maximum of observed values of feature  $A$  respectively.

Another commonly used normalization method is z-score normalization. This is particularly useful when the dataset is expected to contain outliers. Outliers can bias the min-max normalization, because values are scaled between the minimum and maximum values. By applying z-score normalization, new feature values have a mean of 0 and a standard deviation of 1. There is a variation in z-score normalization, which is even more robust to outliers. It works simply by replacing standard deviation with mean absolute deviation. (Gracia et al. 2015, 47-48)

Z-score normalization is applied to observation  $v$  of feature  $A$  between range  $[a, b]$  as follows:

$$v' = \frac{v - \bar{A}}{std(A)}, \quad (2)$$

where  $std(A)$  is the standard deviation, and  $\bar{A}$  is the mean of observed values of feature  $A$  respectively.

### 4.1.2 Dealing with missing values

In the industrial environment, data is usually incomplete, noisy, and inconsistent. It, therefore, requires processing before it can be used in further analysis. In the industrial environment, missing sensor data is very common, and it happens for various reasons. Data may be missing because of unreliable sensors, network communication errors, synchronization problems, and different kinds of equipment failure. An example of an incomplete dataset can be seen in Figure 4-2. (Gracia et al. 2015, 40; Guzel et al. 2019)

		Features					
		1	2	3	4	...	m
Observations	1		?				
	2			?	?		
	3						
	4		?				
	5		?		?		
	...						
	n		?				?

Figure 4-2 Tabular data with MVs. Reproduced from García et al. (2015, 61).

Garcia et al. (2015, 60-61) identify three common approaches for dealing with missing data:

- The simplest way is to discard observations containing missing values (MV). However, this is not usually possible if the number of MVs is substantial. Another concern is that there may be a pattern behind missing values. Important information may be lost if observations with MVs are discarded.
- Another approach is to apply maximum likelihood procedures. A model is built with a complete part of the dataset, and imputation is conducted in the form of sampling.
- The third approach is to use imputation methods, in which MVs are filled with estimated ones. The features are not usually independent of each other. MVs can, therefore, be estimated by identifying relationships between features.

There are different assumptions about missing data. Methods for imputation should be selected based on these assumptions. Common assumptions about missing data are:

- Missing at random (MAR) assumes that the probability that an observation has a missing value for a feature depends on other features rather than the values of the feature itself
- Missing completely at random (MCAR) assumes that the probability that an observation has a missing value for a feature does not depend on the values of the feature itself, nor on other features
- Missing not at random (MNAR) assumes that the probability that an observation has a missing value depends on the feature itself, as well as other features

Numerous imputation methods are available, and the imputing of missing data may be the focus of a thesis in its own right. Due to the scope of this thesis, only a short review of some imputation methods is presented.

In the study conducted by Steiner et al. (2016), MAR was assumed for each dataset. The study compared straightforward imputation methods, such as mean/median imputation, the last observation carried forward (LOCF) method, the simple random imputation to expectation-maximization (EM) algorithm, and multiple imputations (MI) using the Markov Chain Monte Carlo (MCMC) simulation. The study concludes that when EM and MCMC were applied to fill MVs in the data, better prediction results were achieved.

The article by Guzel et al. (2019) attempts to tackle missing sensor data problems by utilizing Deep Learning (DL) and the Adaptive-Network-based Fuzzy Inference System (ANFIS). The study concludes that DL and ANFIS outperform non-linear models used in the study in terms of root-mean-square error. ML methods are becoming popular in missing data estimation according to Guzel et al. (2019). K-nearest neighbor (KNN) is one of the most commonly used algorithms in missing data problems, despite the fact that it was originally introduced as a classification algorithm. Tutz et al. (2015) showed that nearest neighbor methods performed well in a high dimensional setting in which the number of features was high compared to the observations.

Multivariate Imputation methods, such as multivariate imputation by chained equations (MICE), are easy to apply through libraries built for R and Python. MICE take into account the process that created the missing data and preserve the relations within the data and the uncertainty about these relations. MICE work under the assumptions of MAR and MNAR. However, in the case of MNAR, additional modeling assumptions are required which affect the produced imputations. (Van Buuren et al. 2011)

In the empirical part mean, LOCF, and MICE are utilized to estimate missing values in the dataset. MICE is used under the assumption of MAR.

### 4.1.3 Dealing with noise

Another common problem with raw data is noise. Noise can be defined as unwanted data items, irrelevant features, or data points that are not in line with the rest of the records. There are various causes of noise. For example, measuring devices may be malfunctioning, or errors may occur when sending/retrieving data to/from data storage. Noise can reduce system performance in terms of accuracy, model-building time, size, and interpretability. (Zhu et al. 2004; Rathi 2018)

The classification task can be exhaustive, even without noise. Sometimes classes form small disjuncts inside other classes. Classes can also have similar characteristics which lead to overlapping and reduced classification performance. When noise is present in data, it may lead to extreme overlapping, due to irrelevant noisy observations. In Figure 4-3, observations are divided into safe, borderline, and noisy examples. Safe examples are clearly separate from the decision boundary and belong to their own class. Borderline examples are near the decision boundary and are therefore easily misclassified. Noisy examples fall inside the wrong class and cannot be classified correctly. (García et al. 2015, 109)

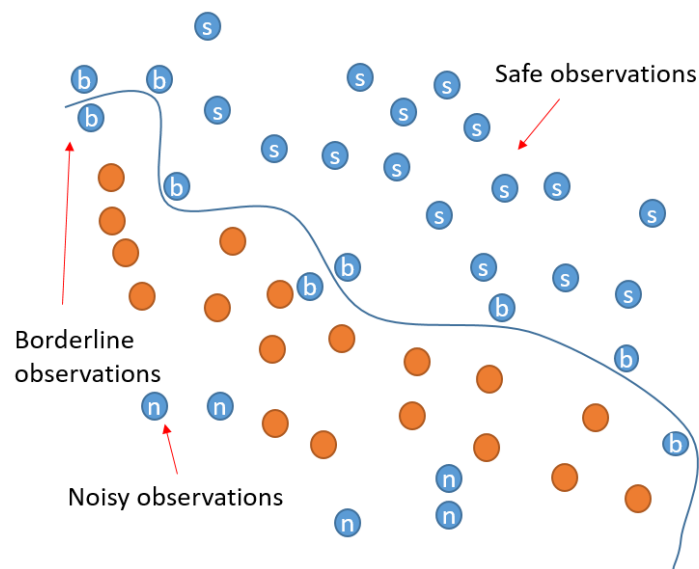


Figure 4-3 Safe, borderline, and noisy observations. Reproduced from García et al. (2015, 110).

There are two types of noise according to García et al. (2015, 110-111):

- Class noise is incorrectly labeled classes, due to data entry errors or lack of knowledge when labeling observations. Class noise can be divided into contradictory examples and misclassifications. Contradictory examples are duplicate examples with different class labels. Misclassifications are observations labeled in the wrong class.
- Feature noise is considered to be invalid feature values and MVs.

Noise can be handled by multiple classifier systems (MCS). MCS aim to gain noise robustness by combining multiple classifiers. MCS reduce the individual problems of each classifier caused by noise. MCS can also be utilized in regression problems. Instead of choosing the best label, the final output is averaged among all models in the MCS. In this thesis, ensemble methods like bagging and boosting are utilized to reduce the influence of noise.

MCS is a parallel approach which means that all available classifiers are given the same input. Outputs are merged with a voting scheme to acquire a final prediction. Sáez et al. (2013) introduce various voting schemes used in classification problems. Two of the methods which can also be applied in regression problems are:

- The majority vote (MAJ) approach, assigns an observation to a class that receives most of the votes among all classifiers.
- A weighted majority vote is a similar approach to MAJ. Labels assigned by each classifier are weighted according to the accuracy of the model in the training phase.

#### 4.1.4 Methods for feature selection

Big data presents new challenges in terms of feature selection, because a number of features in the data can be enormous. Finding the best feature subset from thousands of features can be exhausting. Dimensionality is a serious problem for many ML algorithms. The term “the curse of dimensionality” often appears in the literature. Dimensionality increases computational complexity, which increases training time and decreases model performance (Li et al. 2016). The article by Li et al. (2016) offers an example of relevant, redundant, and irrelevant features. An example can be found in Figure 4-4.

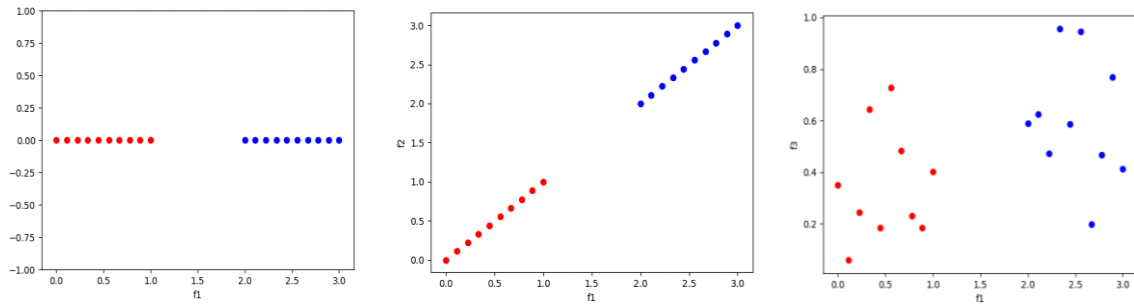


Figure 4-4 Three features  $f_1$  (relevant),  $f_2$  (redundant) and  $f_3$  (irrelevant). Reproduced from Li et al. (2016).

In Figure 4-4, the first feature,  $f_1$ , is relevant, because it can be used to classify data into two classes, blue and red.  $f_2$  is a redundant feature, because it is strongly correlated with  $f_1$  and thus has no additional value in the classification task in hand.  $f_3$  is an irrelevant feature, because both classes exhibit similar behavior regarding  $f_3$ .

Feature selection can be divided into filter, wrapper, and embedded methods. Filter methods are independent of the learning algorithm. They can be used in any situation, but selected features may not be optimal, because there is no learning algorithm guiding the selection of features. Wrapper methods are computationally intensive, because features are evaluated by their contribution to the learning algorithm's predictive performance. Embedded methods are a compromise between filter and wrapper methods. Embedded methods interact with the underlying model. They are more efficient than wrapper methods, because they do not need to iterate through every feature subset. (Li et al. 2016)

Li J. et al. (2016) have made a comprehensive review of feature selection methods for conventional data. Methods are divided into four main categories: similarity-based; information theoretical-based; sparse learning-based; and statistical-based methods.

Similarity-based methods assess feature importance by their ability to approximate similarity within data. Supervised feature selection methods utilize observation labels to assess similarity. Unsupervised methods use various distance metrics. Methods in this family are independent of the learning algorithms. A drawback of these methods is that most of these algorithms cannot handle feature redundancy. It may lead to a subset of highly correlated features. (Li J. et al. 2016)

Information theoretical-based methods aim to minimize redundancy and maximize the relevance of features. Most of these algorithms are supervised, because feature relevance is often assessed by its correlation to class labels. In addition, these algorithms often work only with discrete data. (Li J. et al. 2016)

Sparse learning-based methods have received attention in recent years due to their performance and interpretability. The feature selection of these methods is embedded in the learning algorithms. It can lead to very good performance in a specific learning algorithm. Features thus chosen are not guaranteed to perform well with other learning algorithms. (Li J. et al. 2016)

Statistical-based feature selection methods are often used as filtering methods. They utilize different statistical measures instead of learning algorithms. These methods often analyze features individually, meaning feature redundancy is ignored. Statistical-based feature selection methods are often used in data preprocessing. (Li J. et al. 2016)

Additionally, there are hybrid feature selection, deep learning, and reconstruction-based methods. These methods cannot be classified into the categories mentioned above. The idea in hybrid feature selection methods is to generate subsets of features via different feature selection methods and choose the best features from each of these subsets. Feature selection is usually embedded in the model in deep learning feature selection methods. Relevant features are chosen between the input layer and the first hidden layer. Reconstruction-based methods define a feature's relevance by its ability to describe original data with the reconstruction function. (Li J. et al. 2016)

Some methods that are suitable for regression problems are described below and used in the empirical section of the thesis. These methods are Regression Relief (RReliefF), Least Absolute Shrinkage and Selection Operator (Lasso), Correlation-based feature selection (CFS), Low variance, and Recursive Feature Evaluation (RFE).

#### *RReliefF*

Robnik-Sikonja et al. (2003) propose two algorithms, ReliefF for classification and RReliefF for regression. Both algorithms are supervised similarity-based filter methods. Algorithms are an extension of the original Relief algorithm. The original Relief algorithm works in a supervised fashion and only for binary classification problems. The quality of features is calculated as follows:

$$W[A] = \frac{P(\text{different value of } A \mid \text{nearest observation from different class})}{P(\text{different value of } A \mid \text{nearest observation from same class})} \quad (3)$$

It estimates the quality of features by their ability to separate observations that are near to each other. Robnik-Sikonja et al. (2003) state that ReliefF and RReliefF work in presence of noise and MVs. The ReliefF algorithm works by randomly selecting an observation  $R_i$ , and searches for the nearest neighbors from the same class (nearest hits) and the nearest neighbors from other classes (nearest misses). The quality of features is then based on

feature value, and nearest hits and misses. In regression problems, the nearest hits and misses cannot be calculated. Nearest hits and misses are therefore replaced in RReliefF as follows:

$$P_{diffA} = P(\text{different value of } A \mid \text{nearest instances}) \quad (4)$$

$$P_{diffC} = P(\text{different prediction} \mid \text{nearest instances}) \quad (5)$$

and

$$P_{diffC|diffA} = P(\text{different prediction} \mid \text{different value of } A \text{ and nearest instances}) \quad (6)$$

so  $W[A]$  for regression task is calculated using Bayes' rule:

$$W[A] = \frac{P_{diffC|diffA}}{P_{diffC}} - \frac{(1 - P_{diffC|diffA})P_{diffA}}{1 - P_{diffC}} \quad (7)$$

The pseudocode of RReliefF by Robnik-Sikonja et al. (2003) is presented in Figure 4-5. The inputs are training observations  $x$  and the target value ( $\tau(x)$ ). The output is a vector  $W$  that gives quality for every feature. In the empirical part, all the features which receive  $W[A] > 0$  are selected.

```

1. set all  $N_{dc}$ ,  $N_{dA}[A]$ ,  $N_{dc\&dA}[A]$ ,  $W[A]$  to 0;
2. for  $i := 1$  to  $m$  do begin
3.     randomly select observation  $R_i$ ;
4.     select  $k$  instances  $I_j$  nearest to  $R_i$ ;
5.     for  $j := 1$  to  $k$  do begin
6.          $N_{dc} := N_{dc} + \text{diff}(\tau(I_j), R_i, I_j) \cdot d(i, j)$ ;
7.         for  $A := 1$  to  $a$  do begin
8.              $N_{dA}[A] := N_{dA}[A] + \text{diff}(A, R_i, I_j) \cdot d(i, j)$ ;
9.              $N_{dc\&dA}[A] := N_{dc\&dA}[A] + \text{diff}(\tau(I_j), R_i, I_j) \cdot$ 
10.                 $\text{diff}(A, R_i, I_j) \cdot d(i, j)$ ;
11.         end;
12.     end;
13. end;
14. for  $A := 1$  to  $a$  do
15.      $W[A] := N_{dc\&dA}[A]/N_{dc} - (N_{dA}[A] - N_{dc\&dA}[A])/(m - N_{dc})$ ;

```

Figure 4-5 RReliefF algorithm. Reproduced from Robnik-Sikonja et al. (2003).

In Figure 4-5,  $N_{dc}$ ,  $N_{dA}[A]$ , and  $N_{dc\&dA}[A]$  are the weights for different target values  $\tau(I_j)$  (line 6), different features (line 8), and different predictions and different features (lines 9 and 10) respectively.  $m$  is a user-defined parameter that determines how many times the process is repeated. The term  $d(i, j)$  in Figure 4-5 (lines 6, 8 and 10) is:

$$d(i, j) = \frac{d_1(i, j)}{\sum_{l=1}^k d_1(1, l)} \quad (8)$$

and

$$d_1(i, j) = e^{-\left(\frac{\text{rank}(R_i, I_j)}{\sigma}\right)^2}, \quad (9)$$

where  $\text{rank}(R_i, I_j)$  is the rank of the observation  $I_j$  in a sequence of observation ordered by the distance from  $R_i$ , and  $\sigma$  is a user-defined parameter that controls the influence of the distance.

### Lasso

Lasso is a sparse learning-based embedded method. Lasso was proposed by Tibshirani (1996). Lasso utilizes  $l_1$ -regularization, which limits the power of each coefficient. Some coefficients in the model can be reduced to exactly zero. These features can, therefore, be removed. Tibshirani (1996) defines the lasso estimate  $(\hat{\alpha}, \hat{\beta})$  as follows:

$$(\hat{\alpha}, \hat{\beta}) = \arg \min \left\{ \sum_{i=1}^N (y_i - \alpha - \beta_j x_{ij})^2 \right\} \quad \text{subject to } \sum |\beta_j| \leq t., \quad (10)$$

where  $x_i = (x_{i1}, \dots, x_{ip})^T$  is the feature vector of  $i$ :th observation,  $y_i$  is the corresponding target,  $N$  is the number of observations,  $t$  is the tuning parameter,  $\hat{\beta} = (\hat{\beta}_1, \dots, \hat{\beta}_p)^T$ , and  $\alpha$  is  $\hat{\alpha} = \bar{y}$ . In the empirical part, all features assigned with a non-zero coefficient are selected for the “optimal” feature subset.

### CFS

CFS is a supervised statistical-based filter method. CFS uses correlation-based heuristics in the evaluation of a feature subset. CFS attempts to maximize the correlation between the target feature and the feature subset while minimizing the correlation between features in the feature subset. Finding the optimal feature subset this way is computationally challenging. CFS tackles this issue by calculating the utility of each feature. It considers feature-target and feature-feature correlation. It then starts with an empty set and expands it one feature at a time. Addition order for features is determined by utility. The addition continues until some stopping criteria are met. (Li J. et al. 2016)

The feature subset is evaluated using the following function, first introduced by (Ghiselli, 1964):

$$CFS\_score(S) = \frac{k\bar{r}_{cf}}{\sqrt{k+k(k-1)\bar{r}_{ff}}}, \quad (11)$$

where the CFS score describes the quality of the feature subset  $S$  with  $k$  features.  $\overline{r_{cf}}$  is the average target-feature correlation, and  $\overline{r_{ff}}$  is the average feature-feature correlation in the feature subset  $S$ . The numerator can be seen as a measure of how well  $S$  describes the target and the denominator as the measure for redundancy within  $S$ . (Hall et al. 1999)

#### *Low variance*

Low variance is a statistical-based filter method. Low variance features contain less information than features with higher variance. By using this method, all features are eliminated which have lower variance than the predefined variance threshold. All features with zero variances should be removed, because they do not contain any information. A low variance method is commonly used as a preprocessing step rather than as an actual feature selection method. (Li J. et al. 2016)

#### *RFE*

RFE is a supervised wrapper method. The ranking of features differs, depending on the learning algorithm in use. In the scikit-learn package, features are ranked by the coefficients of features or feature importance metric. In this thesis, RFE ranks features based on their coefficients, because the learning algorithm used in the feature selection is linear regression. The higher coefficient value indicates the greater importance of that feature. RFE returns the user-specified number of highest ranked features. One must iterate through a number of features to acquire the feature subset which produces the best accuracy. (Scikit-learn 2019; Guyon et al. 2002)

The steps through RFE are described in the pseudocode in Figure 4-6.

1. divide data into training and testing sets;
2. **for**  $i:= 1$  **to** the maximum number of features **do**
  - a. train model with the training set containing all the features;
  - b. select  $i$  features with largest coefficients or feature importance;
  - c. save selected feature subset;
  - d. save accuracy with the testing set;
3. choose feature subset which produced the best accuracy with the testing set;
4. **end**;

*Figure 4-6 Pseudocode for RFE*

## 4.2 Machine learning models

In this chapter, some ML techniques for regression problems are briefly reviewed.

Traditionally, models are physics-based. This means that the relationships between features are explained by the laws of physics. This approach requires extensive knowledge of the process in hand. Processes may have so many features that deriving an accurate model is very complicated. ML techniques are one way to overcome this problem if a lot of data is available. Learning algorithms can learn relationships between features by fitting a curve to the training data. It is an iterative process, which aims to minimize the error between the fitted curve and data points. (Mehrotra et al. 2017, 57-58)

### 4.2.1 Linear regression

Linear regression can be used to model continuous features, such as electricity consumption. The method assumes that features have linear relationships. When the term “linear regression” is used in the literature, it usually encompasses multiple linear regression as well. (Ryan 2009, 146)

The function of linear regression is

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_m X_m, \quad (12)$$

where  $Y$  is model output,  $X_i, 1, \dots, m$  is an independent feature, and  $\beta_i, 0, \dots, m$  is the corresponding coefficient. The goal is to minimize the difference between model outputs and observed values by optimizing coefficients as known as least square estimates.

Ryan (2009, 133-135) illustrates how matrix algebra can be applied to regression. Least square estimates for function

$$Y = X\beta + \varepsilon \quad (13)$$

can be obtained by using the function

$$\beta^\wedge = (X'X)^{-1}X'Y, \quad (14)$$

where

$$X'X = \begin{bmatrix} n & \sum X \\ \sum X & \sum X^2 \end{bmatrix}, \quad (15)$$

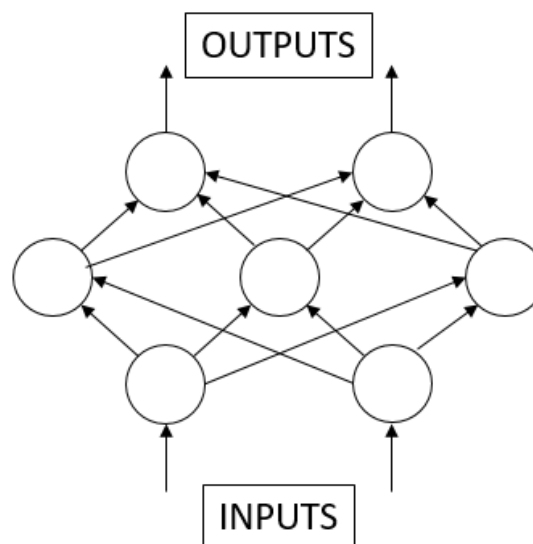
and

$$X'Y = \begin{bmatrix} \sum Y \\ \sum XY \end{bmatrix}. \quad (16)$$

Linear regression may also be solved with a gradient descent method. Many algorithms work iteratively to find these optimal coefficients. These processes are usually gradient solvers. The gradient descent methods work by changing coefficients on every iteration toward a better fit. Coefficients are changed until the average error between observed values and predicted values do not change, or the maximum number of iterations is reached. (Rebala et al. 2019, 27-36)

#### 4.2.2 Multilayer perceptron

A multilayer perceptron is an artificial neural network. It can be used for classification and regression problems. A three-layer neural network can be seen in Figure 4-7.



*Figure 4-7 Three-layer neural network. Reproduced from (Krawczak 2013, 3).*

The network consists of neurons, which are the individual processing units of their inputs. The neurons are linked by connections, and each connection has a weight. Neurons are in the form of layers, and information moves through the network layer to the next layer. Each neuron of each layer receives information from each neuron from the previous layer. The first layer receives features as inputs, and the layers after the first layer receive inputs from the previous layer. The final layer produces the outputs of the neural network. The operation of a single neuron is illustrated in Figure 4-8. (Krawczak 2013, 1-3)

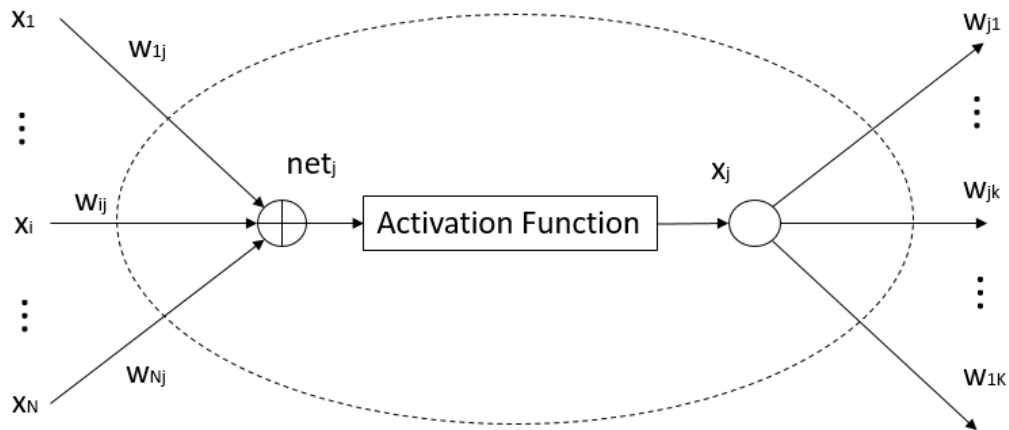


Figure 4-8 Single neuron. Reproduced from (Krawczak 2013, 3).

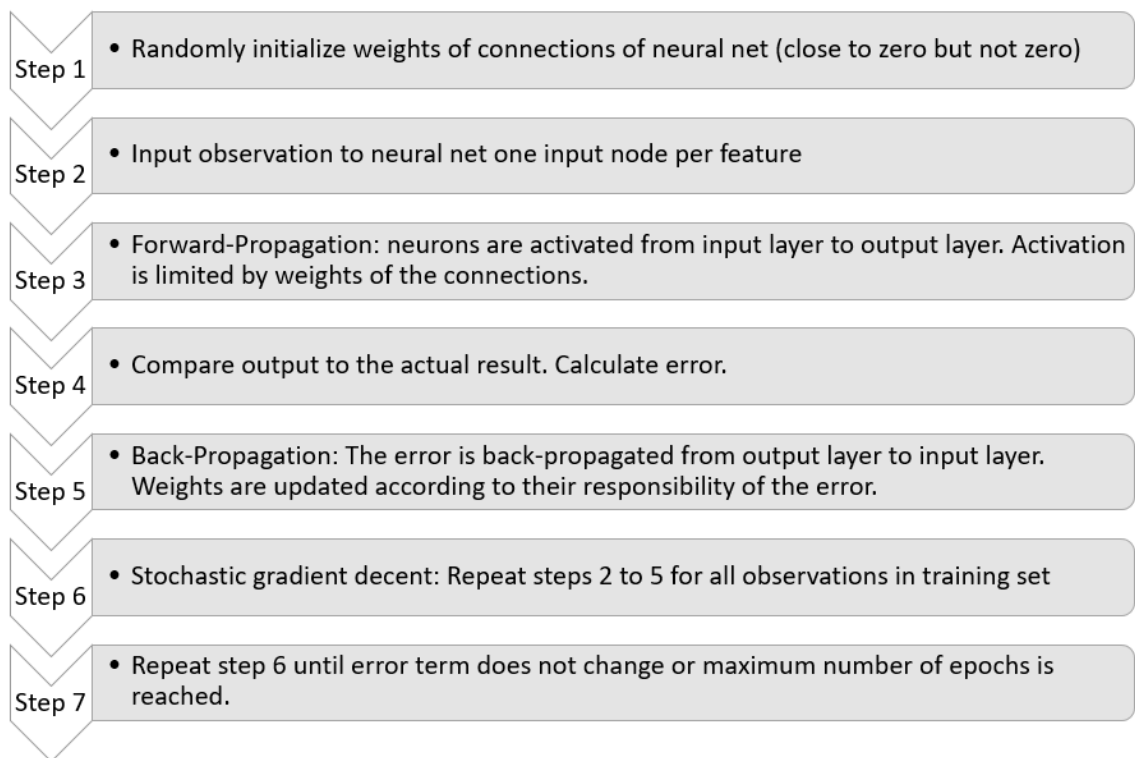
In Figure 4-8, external signals are denoted by  $x_i$ , where  $i = 1, 2, \dots, N$ ,  $x_i$  can be the input to the network or the output of a neuron from the previous layer. Weights of connection are denoted by  $w_{ij}$ , where  $i = 1, 2, \dots, N$  is the index of the incoming signal, and  $j$  is the index of the considered neuron. At the point where weights and  $\theta_j$  are connected, the following calculation is performed:

$$net_j = w_{1j} * x_1 + \theta_1 + \dots + w_{ij} * x_i + \theta_j, \quad (17)$$

where  $\theta_j$  is a bias weight. The activation function determines the value passed to the neurons of the next layer. Rectified activation functions known as Rectified Linear Units (ReLUs) are now a commonly used activation function. ReLUs are simple and fast to execute, alleviate the vanishing gradient problem, and induce sparseness. The rectified linear function is defined as:

$$f_{ReLU}(x_i) = \max(0, x_i). \quad (18)$$

An issue with ReLUs is that negative inputs are always set to zero. This means negative gradient values cannot get past that neuron during back-propagation. Back-propagation is the training method for the neural network, which is done by adjusting the weights of the connections by minimizing errors between the output and actual value. (Godin et al. 2017; Krawczak 2013, 3-4; Rumelhart et al. 1986)



*Figure 4-9 ANN training process.*

The ANN training process is illustrated in Figure 4-9. Stochastic gradient descent is the process where all observations in the training set are inputted to ANN one at a time. The weights of the connections are updated after each iteration. Repeating the process for the whole training set is referred to as an epoch. The required number of epochs depends on the size of the network, learning rate, and size of the training data. The learning rate defines how much weights are updated after each iteration. (Nielsen 2015 15-24; 40-50)

### 4.2.3 Bagging

Bagging methods work by building several estimators for the same prediction task on random bootstrap subsets of the original dataset. The word “bagging” is an acronym of the words “bootstrap aggregating”. More about the bootstrap can be read in the paper by Efron et al. (1994). Bagging is used to reduce the variance of a base estimator. For example, the base estimator can be a decision tree. (Breiman 1996)

Bagging can be used for regression and classification problems. In classification problems, the final prediction is decided using a voting scheme. In regression problems, the final prediction is the average of all individual estimators. Bagging is a relatively easy method to

increase the accuracy of a single learning algorithm. The only downside of this procedure is that interpretability decreases. (Breiman 1996)

#### 4.2.4 Boosting

Boosting is seen as one of the most powerful recently discovered learning ideas. Boosting is originally designed for classification problems, but it was later extended to regression problems. In boosting, an ensemble of weak learners are built sequentially, rather than in parallel as in bagging. For example, an algorithm called the “Adaboost.M1” iteration starts by fitting a decision tree to the data, and all observations have equal weight. After each successive iteration, the observations that are most wrongly estimated (regression) or misclassified (classification) receive higher weights, which forces the following iteration to focus on these observations. More weight is given to accurate learners in an ensemble. A visualized example of Adaboost.M1 algorithm can be seen in Figure 4-10. (Hastie 2009, 387-338)

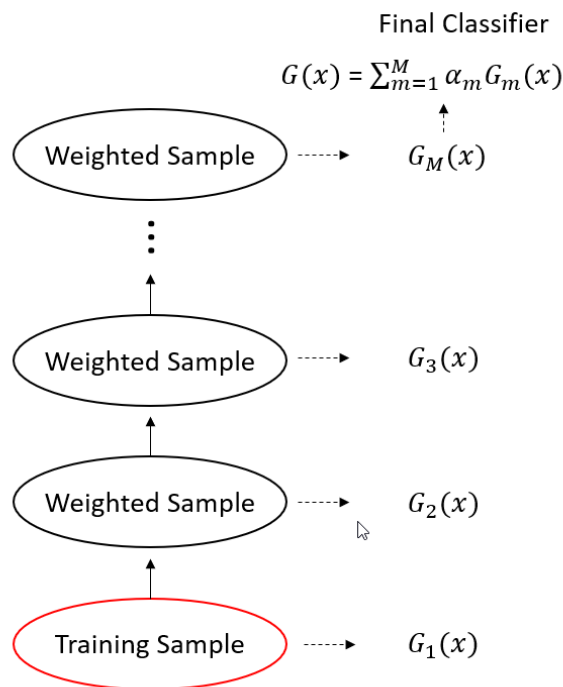


Figure 4-10 Adaboost algorithm. Reproduced from Hastie et al. (2008, 338).

In Figure 4-10, weak learners  $G_m(x), m = 1, 2, \dots, M$ . form an ensemble  $G(x)$  in which each  $G_m(x)$  is weighted by  $\alpha_1, \alpha_2, \dots, \alpha_M$ . The gradient-boosting algorithm works similarly to the Adaboost.M1 algorithm. While Adaboost.M1 identifies wrongly classified/estimated observations by using weights, gradient boosting uses gradients in the loss function. The

loss function is a measure of how well models fit the data. Learning algorithms always seek to minimize the loss function. (Singh 2018)

The hyperparameters of the gradient-boosting algorithm can be tuned to improve the performance of the model. Hyperparameters are parameters that affect the learning process of ML learning algorithms. Hyperparameters are independent of data, unlike parameters, which are optimized during the training. In this thesis, the focus is on hyperparameters, which have the most influence in the model performance. These hyperparameters are the learning rate (“learning\_rate”), the number of learners (“n\_estimators”), the maximum depth of a tree (“max\_depth”), and the minimum number of samples required to split an internal node (“min\_samples\_split”). More of these hyperparameters are presented in Scikit-learn (2019). The effects of each hyperparameter tuned in this thesis are described as follows:

- learning rate: The contribution of each successive weak learner is decreased by the learning rate. Increasing the learning rate gives more influence to weak learners trained at the beginning of the iteration, and vice versa.
- n\_estimators: The number of learners trained. Usually, when increasing the number of trees, the learning rate is decreased.
- max\_depth: The maximum depth of the individual weak tree learner. The best value depends on the interaction between the features.
- min\_samples\_split: Increasing the number of samples required in a split of an internal node may help to reduce overfitting.

### 4.3 Model interpretation

It can be extremely difficult to interpret the decision making of complex ML models. However, it is still sometimes necessary for various reasons – for example, legal. Complex models are often avoided, because simpler models are more interpretable. Even when accuracy of a complex model is higher than the accuracy of a simple model. To address this problem, Lundberg et al. (2017) proposed the SHAP (Shapley Additive exPlanations) method. SHAP uses an explanation model which is an interpretable approximation of the original complex model.

The basic idea behind additive feature attribution methods is to explain the prediction of the model  $f$  by the explanation model  $g$ . Explanation models often use simplified inputs of the original inputs,  $x$  denoted as  $x'$ . The original inputs are mapped with the mapping function

$$x = h_x(x'). \quad (19)$$

Many of the additive feature attribution methods use an explanation model as follows:

$$g(z') = \phi_0 + \sum_{i=1}^M \phi_i z_i', \quad (20)$$

where  $z' \in \{0,1\}^M$ ,  $M$  is the number of simplified input features, and  $\phi_i$  represents the effect of a  $i$ :th feature,  $\phi_i \in \mathbb{R}$ . The sum of the effect of all features is equal to the original model output  $f(x)$ . (Lundberg et al. 2017)

Lundberg et al. 2017 review the existing additive feature attribution methods that inspired them to develop the SHAP method. The methods are Local Interpretable Model-agnostic Explanations (LIME) (Ribeiro et al. 2016), Shapley sampling values (Strumbelj et al. 2014), Deep Learning Important FeaTures (DeepLIFT) (Shrikumar et al. 2017), Quantitative input influence (Datta et al. 2016), Layer-wise relevance propagation (Bach et al. 2015), and Shapley regression values. (Lipovetsky et al. 2001)

Lundberg et al. (2017) introduce three desired properties of additive feature attribution methods which some of the previously mentioned methods fail to fulfill.

- Local accuracy.

$$f(x) = g(x') = \phi_0 + \sum_{i=1}^M \phi_i x_i' \quad (21)$$

The output of explanation model  $g(x')$  should match the output of the original model  $f(x)$  when  $x = h_x(x')$

- Missingness.

$$x_i' = 0 \rightarrow \phi_i = 0 \quad (22)$$

Features missing from the original input have no effect.

- Consistency.

Let  $f_x(z') = f(h_x(z'))$  and  $z' \setminus i$  denote setting  $z_i' = 0$ . For any two models  $f$  and  $f'$ , if

$$f'_x(z') - f'_x(z' \setminus i) \geq f_x(z') - f_x(z' \setminus i) \quad (23)$$

for all inputs  $z' \in \{0,1\}^M$ , then  $\phi_i(f', x) \geq \phi_i(f, x)$ .

$$\phi_i(f, x) = \sum_{i=1}^M \frac{\text{card}(z')!(M - \text{card}(z') - 1)!}{M!} [(z') - f_x(z' \setminus i)], \quad (24)$$

where  $\text{card}(z')$  is the cardinality of  $z'$ , and  $z' \subseteq x'$  represents all  $z'$  vectors in which the non-zero entries are a subset of the non-zero entries in  $x'$ .

Function (24) is from combined cooperative game theory results, where  $\phi_i$  are Shapley values. The Shapley value was first introduced by Shapley (1952), who wished to calculate each player's contribution in a coalition game. To meet these three desire properties, Lundberg et al. (2017) propose SHAP values for a unified measure of feature importance.

SHAP values are easily interpreted. Features' SHAP values are added to the base value of the target feature to obtain the model output. A single SHAP value represents the absolute contribution of that feature. An example of how features contribute to the prediction is presented in Figure 4-11.

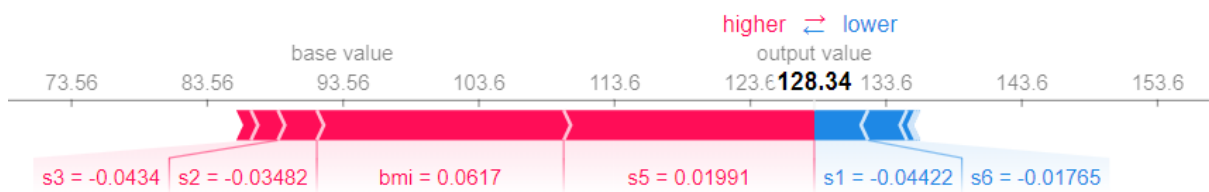


Figure 4-11 Force plot (Feature contribution in a single prediction)

Figure 4-11 is produced with the diabetes data provided in the SHAP Python library. Figure 4-11 shows each feature's impact on the output value. With zero features in the model, the prediction is the base value, which is 93.56, in Figure 4-11. The size of the bar represents the SHAP value, and the corresponding number is the feature value. The SHAP values of each feature in Figure 4-11 can be seen as a horizontal bar plot in Figure 4-12.

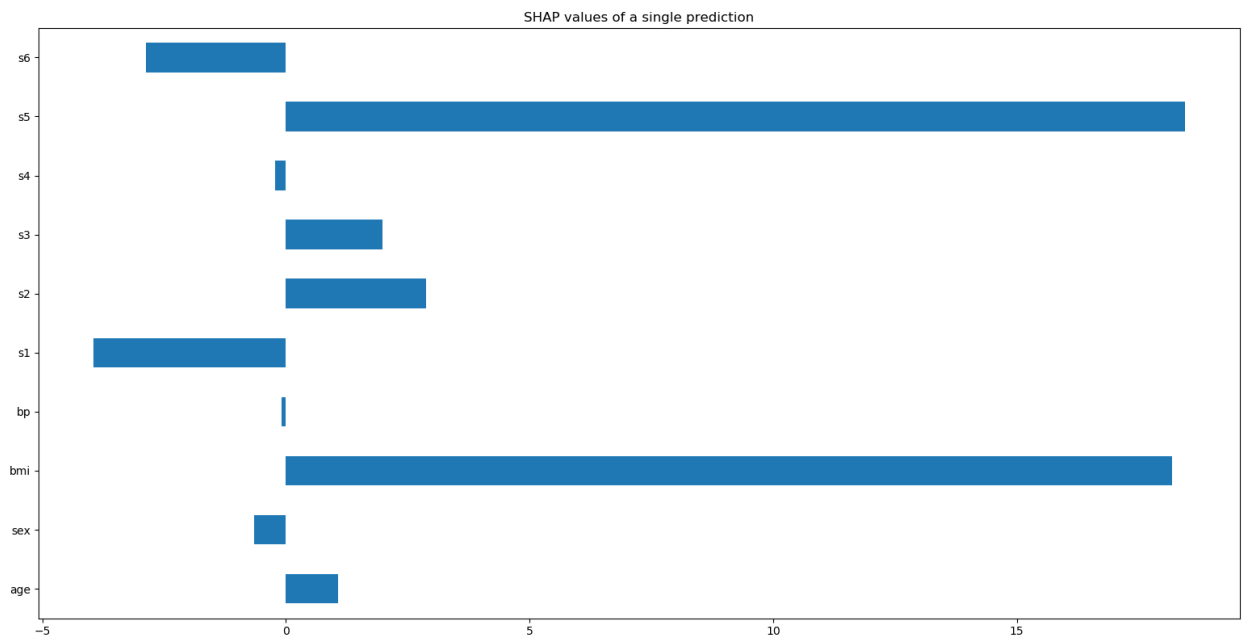


Figure 4-12 SHAP values of features

Features with positive SHAP values have a positive impact. Features with negative SHAP values have a negative impact on the model output. In Figure 4-12, it is easy to see that “bmi” and “s5” have the most positive impact on the model, while “s6” and “s1” decrease the model output. The SHAP values of “bmi” with different feature values can be seen in the dependence plot in Figure 4-13.

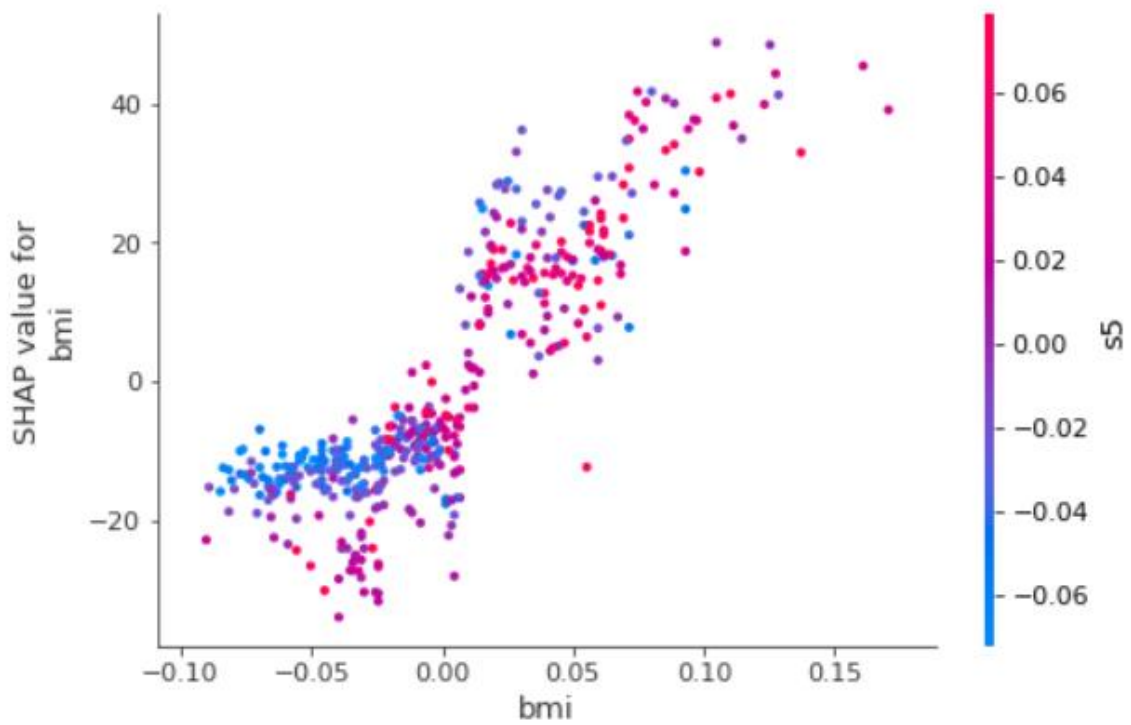


Figure 4-13 Dependence plot (SHAP values of a single feature)

Figure 4-13 shows that a lower “bmi” decreases the model output. The coloring of the data points is derived from the feature values of “s5”. The colors indicate that the majority with a low “bmi” have a low “s5”. In Figure 4-14, the SHAP values of each feature for the whole dataset, which is called a summary plot, can be seen.

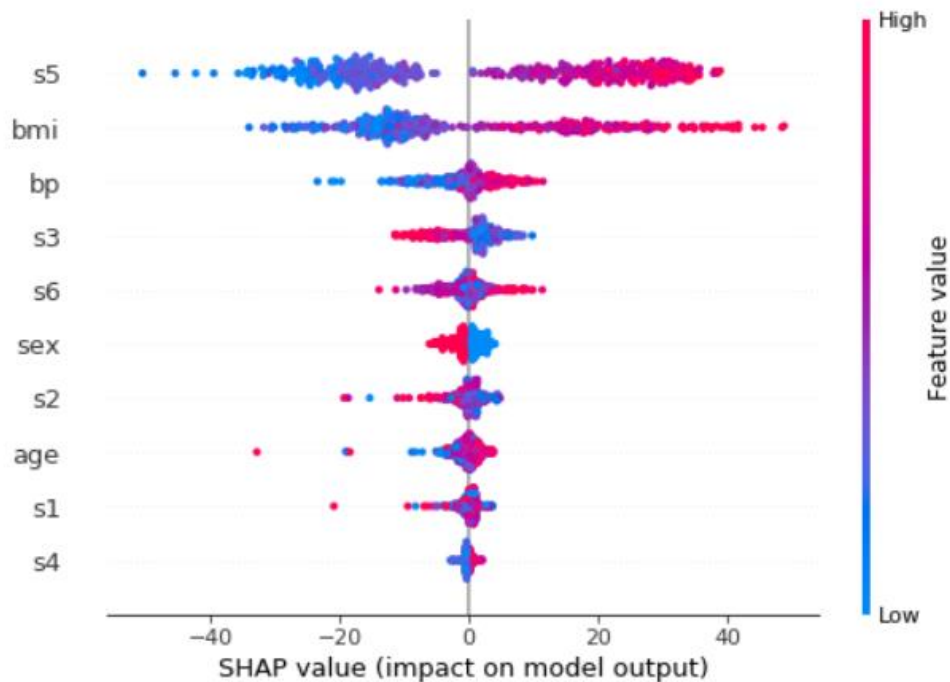


Figure 4-14 SHAP summary plot.

In Figure 4-14, the features are on the vertical axis, and the SHAP value is on the horizontal axis. The coloring represents feature value. Most of the features have a positive correlation with the target. When features receive low values (blue), the corresponding SHAP values are negative, and vice versa. There are some exceptions, “s3”, for example. High “s3” values have a negative impact on the model output, and vice versa.

The SHAP method can be used for interpreting complex models. The force plot enables an interpretation of single predictions, the dependence plot helps in understanding the impact of a feature, and the summary plot gives an overview of how a model utilizes features.

#### 4.4 Root cause analysis

The purpose of RCA is to resolve what caused the problem in the first place. Usually, problems are solved with ad hoc solutions that do not remove the root cause, and the problem reappears. RCA can be an exhaustive process that requires the right personnel to investigate the problem. Humans have a tendency to assume things based on their previous experiences. Assumptions may help in finding a root cause but may also prohibit it if facts are disregarded. Assumptions that cannot be proven should, therefore, be disregarded. (Okes 2009; Mobley 1999)

Okes (2009) lists five different approaches to RCA:

- Events and causal factory analysis – widely used for major single event incidents such as explosions.
- Change analysis – used when system performance changes significantly. It assesses all the available information that might have led to performance change.
- Barrier analysis – focuses on what controls are used in the process to either prevent or detect a problem, and what is in danger of failing.
- Management oversight and risk tree analysis – utilizes a tree diagram to discover what occurred, and why.
- Kepner-Tregoe Problem Solving and Decision Making – This model provides four phases to resolve problems: situation analysis; problem analysis; solution analysis; and potential problem analysis

The approach in this thesis is closest to change analysis, and data plays a central role. Significant performance changes are examined by calculating the Pearson correlation with error and measured features that are not included in the model. Cross-correlation may be utilized to capture process features that have delayed the impact on performance change.

Cross-correlation is calculated between two vectors. It works by moving another vector forward or backward in time within a time window. When moving the vector relative to another vector, both vectors must have the same number of observations. The interval of time between the first observations of each vector is the lag or offset. (Boker et al. 2002)

Boker et al. (2002) have an excellent example of cross-correlation. Cross-correlation is calculated between time series,  $X = \{X_1, \dots, X_N\}$  and  $Y = \{Y_1, \dots, Y_N\}$  with equal intervals of time,  $t$ , between observations.  $Y$  is lagged by a positive lag of  $\tau$ . The function can be defined as follows:

$$r(X, Y, \tau) = \frac{1}{N-\tau} \sum_{i=1}^{N-\tau} \frac{(x_i - \bar{X})(y_{i+\tau} - \bar{Y})}{std(X)std(Y)}, \quad (25)$$

where  $\bar{X}$  and  $\bar{Y}$  are the grand means, and  $std(X)$  and  $std(Y)$  are the standard deviations of  $X$  and  $Y$  respectively. The function presents the Pearson correlation between the two time series, lagged by  $\tau$  observations.

## 5 Proof of concept

In this chapter, the theoretical framework is utilized to attain a robust solution for anomaly detection and data-driven RCA. First, methods from the theoretical framework are applied with dryer group 3 (Chapter 2) data. The target feature in modeling is the total electricity consumption of dryer group 3 drives. This chapter is divided into 7 subchapters, which present each step in the process of creating a model and conducting data-driven RCA.

The VII platform (Chapter 3) is utilized in data acquisition. The VII data pipeline uploads new data to SF continuously in the conventional form for further processing and analysis. SF API provides easy tools to access the data from any integrated development environment (IDE). The IDE used in the empirical part is Spyder, which is specially designed by and for scientists, engineers, and data analysts (Spyder 2019). The programming language used in Spyder is Python. Many useful external open-source libraries built for Python are used in this study. Some of the most common libraries used in analytical work are:

- Pandas provides tools for data processing, formatting, and simple data analysis (Pandas 2019).
- NumPy is designed for scientific computing and data formatting (Numpy 2019).
- Scikit-learn is a simple and efficient tool for data mining and data analysis, especially for ML (Scikit-learn 2019).
- Matplotlib is a library for plotting data and creating interactive visualizations (Matplotlib 2019).

These four widely known Python libraries handle most of the tasks addressed by the study. However, there may be tasks that require methods that are not included in these libraries. The theoretical background of data processing, modeling, and data-driven RCA are presented in Chapter 4. **In the empirical part, features are referred to as tags**, because all features are measurements from PMs where measurements are called tags.

### 5.1 Data collection

Data is downloaded from SF using SF API. Data goes through the pipeline presented in Chapter 3 before it is available in SF. Data in snowflake is divided into fact tables and dimension tables. Fact tables contain tag values and timestamps. Dimension tables contain descriptive information of the tag – for example, its identification code or a “tag name” and unit of measurement.

Information from fact and dimension tables is combined with a join query to designate tag values with the correct descriptive information. The time between measured values changes from a few seconds to almost a minute. To make frequency consistent, an SQL function called “DATE\_TRUNC” can be used. Thousands of different tags are available, but not all are relevant when creating a model. Tags specifically relevant for the model are determined with SMEs and included in the SQL query. After constructing the SQL query, it may be run in SF UI or with SF API to fetch the data. In this thesis, SF API is used, because it enables larger datasets to be downloaded to IDE and saved to local storage. Downloaded data is received in a list of dictionaries, in which each dictionary corresponds to an observation of a tag. The relevant information of which each observation consists are a timestamp, a tag name, and a tag value. The timestamp refers to the time when the tag value is measured, the tag value is the measured value, and the tag name is the measurement’s identification code. The SQL query used to fetch data is given in Appendix I.

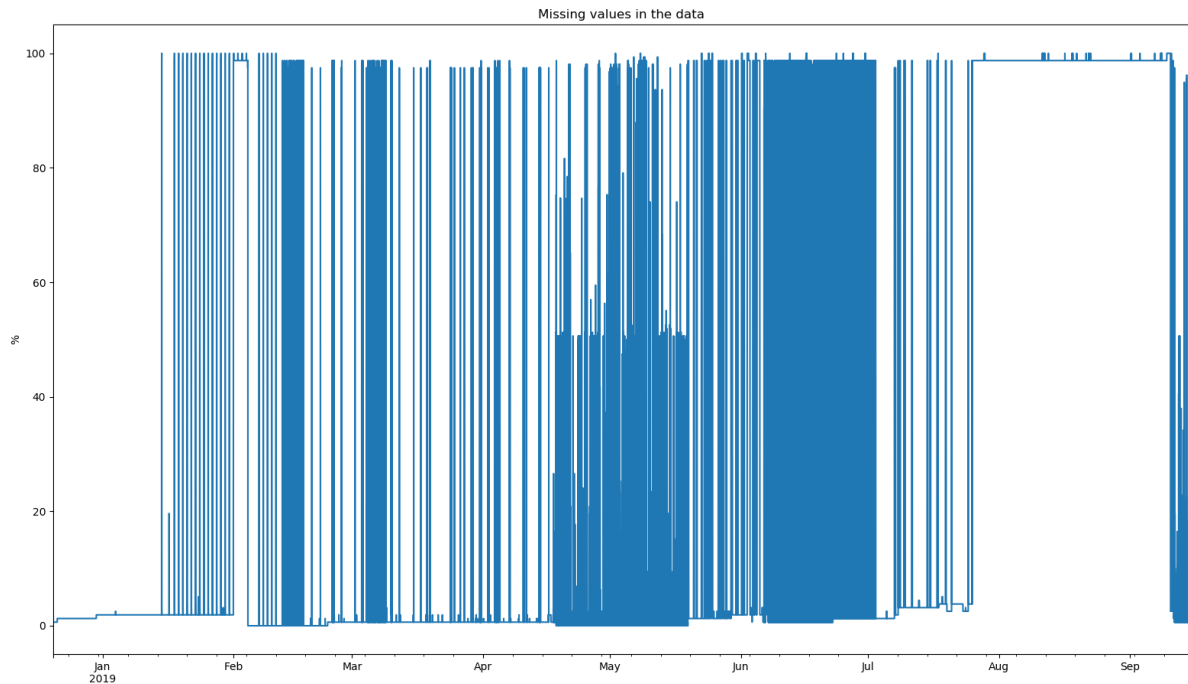
## 5.2 Data processing

A list of strings is not yet the most convenient way to use data. Data is transformed into the dataframe where it is in tabular form, timestamp as an index, and tag names in columns. When data is in this form, it can easily be used in many algorithms from various libraries. As stated in Chapter 4.1.2, data is not often perfect in an industrial environment, so an exploratory analysis of the dataset is in place. The properties of the dataset are provided in Table 5-1.

*Table 5-1 Properties of the dataset.*

<b>ROWS (TIMESTAMP)</b>	386,866
<b>TAGS</b>	158
<b>FIRST OBSERVATION</b>	2018-12-20 08:15:00
<b>LAST OBSERVATION</b>	2019-09-15 00:00:00
<b>RESOLUTION</b>	1 min

Table 5-1 shows that there are fewer rows than minutes between the first and last observations. This means there are MVs in the data which prohibit the use of many algorithms. Figure 5-1 shows the percentage of the tags missing in the function of time.

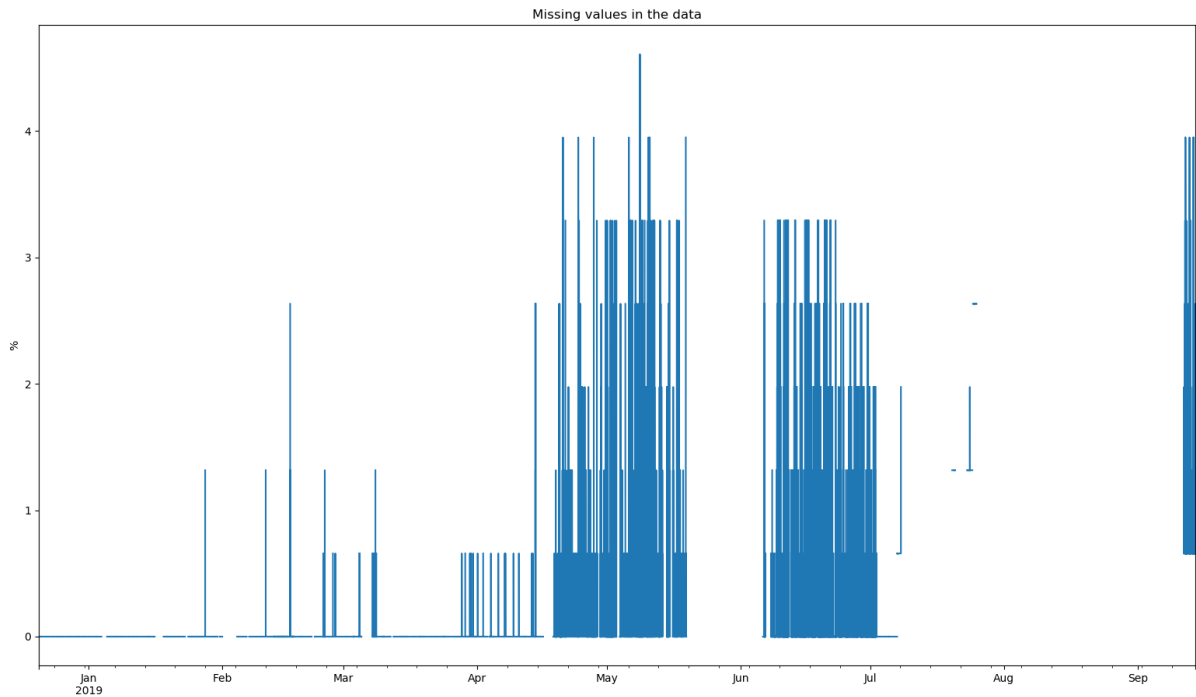


*Figure 5-1 Missing values in the function of time.*

In Figure 5-1, the x-axis corresponds to the time, and the y-axis corresponds to the relative number of missing values. The dataset clearly has many missing values, and at some timestamps, all the tag values are missing. The number of missing values is so high that imputation methods cannot be used immediately. The best accuracy from imputation methods is achieved when most of the data is known. It is necessary to delete observations and tags which have too many missing values before applying imputation methods.

Elimination is done by following these steps:

1. Remove observation if the target feature (total power of dryer group 3) is missing.
2. Remove observation if sheet break information is missing. Sheet break indicates when PM is running and when there is a break.
3. Remove observation if more than 5 percent of tags are missing. 5 percent = 8 tags.
4. Remove tag if more than 5 percent of its observations are missing.
5. Remove observation if sheet break is 1, which means that PM is not running, because PM runs abnormally before and after sheet breaks. The period removed is extended by 30 minutes before and 60 minutes after a sheet break has been 1.



*Figure 5-2 Missing values in the function of time after elimination.*

The number of missing values is greatly reduced after elimination, as can be seen in Figure 5-2. Long periods are removed from the data, because too many tags or observations are missing. For example, no data is available in August. The properties of the new dataset can be seen in Table 5-2.

*Table 5-2 Properties of the dataset after elimination.*

<b>ROWS (TIMESTAMP)</b>	164,720
<b>TAGS</b>	152
<b>FIRST OBSERVATION</b>	2018-12-20 10:04:00
<b>LAST OBSERVATION</b>	2019-07-07 06:14:00

Elimination has reduced observations significantly. Furthermore, few tags were deleted because they contained too many MVs. However, 50 of the tags contained MVs which had to be dealt with. Now each tag and row have at most 5 percent of MVs. A comparison of the imputation methods (see Chapter 4) is made by creating linear regression models for datasets imputed with different imputation methods. The best imputation method is chosen based on the R-squared value of the model on unseen data (test set). Additionally, one model is created with tags that do not have any missing values to benchmark imputation methods. Data observations are randomly divided into train and test sets. Seventy percent of

observations in the data are in the train set, and 30 percent are left in the test set. The results of the comparison can be seen in Table 5-3.

Table 5-3 R-squared and MSE of the test and train set with different imputation methods

METRIC/ METHOD	R2 (TEST)	R2 (TRAIN)	MSE (TEST)	MSE (TRAIN)
<b>BENCHMARK</b>	0.979	0.979	6.453	6.451
<b>MEAN</b>	0.939	0.984	18.428	4.674
<b>LOCF</b>	0.983	0.986	5.04	4.249
<b>MICE</b>	0.986	0.986	4.325	4.230

Table 5-3 shows that there is little difference between R-squared when using different imputation methods. Only a mean imputation leads to a less accurate solution than simply removing the tags that contain missing values. LOCF is almost as accurate as MICE. A tag with most MVs and the different imputations for the tag are visualized in Figure 5-3.

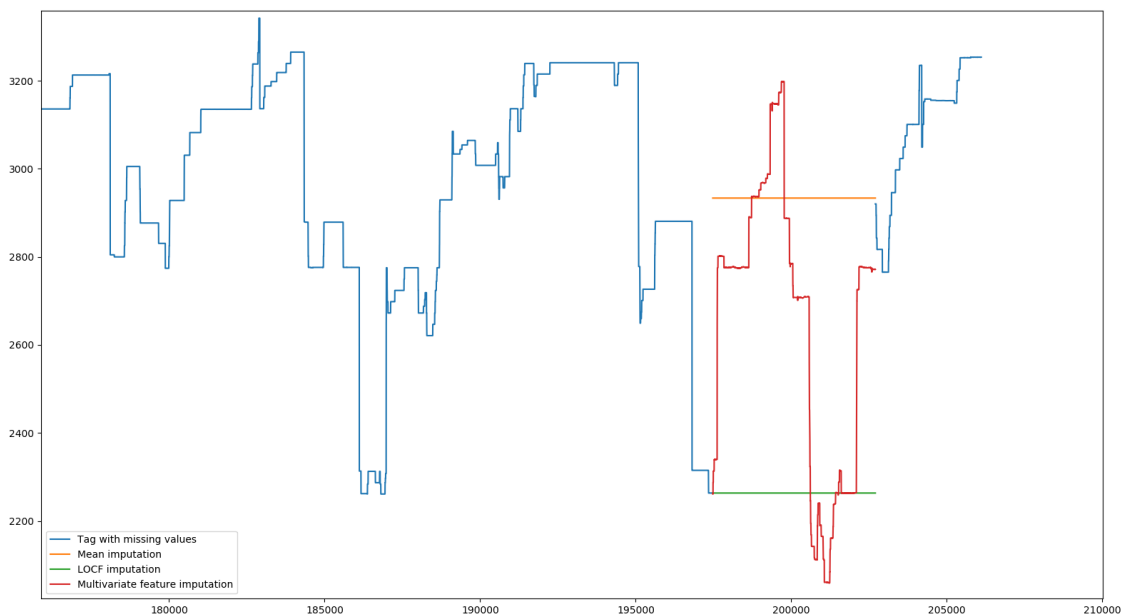


Figure 5-3 Comparison of imputation methods.

While R-squared and MSE of simple imputation methods seem sufficient, the reality may be different, as Figure 5-3 shows. These imputation methods can ruin a dataset if tags have successive MVs. MICE is credible under the assumption of MAR, even when there are successive MVs.

The next step is to remove all the tags which have zero variance, which means they contain no information. Removing features with low variance is also a feature selection method. However, removing zero variance tags can also be used as a preprocessing step. Fifteen tags have zero variance, and after the elimination, the dataset contains 137 tags, including the target feature. Finally, all tags are normalized to have a zero mean and unit variance, using z-score normalization.

### 5.3 Feature selection

In this phase, the goal is to find a feature subset that produces the best results with the linear regression model. Feature selection methods are described in Chapter 4.1.4. Feature sets containing all 136 tags are used as a benchmark for chosen feature selection methods. The feature set is optimized with the smaller subset “feature selection set” of the observations to keep computation time reasonable. The feature selection set consists of 1,000 randomly chosen observations from the processed data. The chosen tags are used to train a linear regression model with complete data divided into train and test sets. More information about this can be found in Table 5-4. Data is divided into train and test sets by date, not randomly as in the previous chapter to enable the results to be easily visualized.

*Table 5-4 Train and test sets, and feature selection set.*

<b>DATASET</b>	<b>START</b>	<b>END</b>	<b>NUMBER OF OBSERVATIONS</b>
<b>FEATURE SELECTION SET</b>	2018-12-20 10:04:00	2019-07-07 06:14:00	1,000
<b>TRAIN SET</b>	2018-12-20 10:04:00	2019-05-19 09:59:00	123,177
<b>TEST SET</b>	2019-06-06 03:48:00	2019-07-07 06:14:00	22,048

Feature selection methods are used with default hyperparameters. The number of feature subsets grows exponentially if methods are used with various hyperparameters.

Table 5-5 R-squared and MSE of the test and train set with different subsets.

METRIC/ METHOD	R2 (TEST)	R2 (TRAIN)	MSE (TEST)	MSE (TRAIN)	TAGS SELECTED
ALL INCLUDED	0.853	0.986	41.356	3.822	137
RRELIEFF	0.935	0.985	18.406	4.139	114
CFS	-0.19	0.11	336.467	241.948	6
RFE	0.9343	0.984	18.474	4.139	112
LASSO	-3.28	0.865	1204.961	16.299	13

Results from the linear regression for each tag subset can be seen in Table 5-5. The low number of tags in the subset leads to much worse performance than the high number of tags. This indicates that most of the tags are important. Including all tags also seems to produce decent results, but it is not optimal. RReliefF and RFE produce the best tag subsets. The tag subset produced by RReliefF is chosen for model building, because it performed best with the test set.

#### 5.4 Model selection

In this chapter, several regression models are applied to the processed dataset. The models used are introduced in Chapter 4.2. The best model is decided based on metrics calculated from the test set. Data is divided in the same way as in the previous chapter. However, some of the observations have to be removed, because the results of the physics-based model were unavailable for the whole period. To make the results more interpretable, the mean absolute error (MAE) is used instead of MSE. Furthermore, 5-fold cross-validation is applied to the train set. Folds are random splits into train and validation sets. The test set is about one month of continuous data. Table 5-6 shows the data division.

Table 5-6 Train and test sets.

DATASET	START	END	NUMBER OF OBSERVATIONS
TRAIN SET	2018-12-31 23:09:00	2019-05-19 09:59:00	123,177
TEST SET	2019-06-06 03:48:00	2019-06-30 23:56:00	22,048

The physics-based model developed at Valmet is used as a benchmark for regression models. The models built in this chapter are linear regression, gradient boosting, bagging, Adaboost, and MLPR. The models are used with default hyperparameters set by the scikit-learn Python library. The best model is chosen for hyperparameter tuning and anomaly detection. The results can be seen in Table 5-7.

*Table 5-7 Performance of different models.*

<b>METRIC/ METHOD</b>	<b>R-2 (TEST)</b>	<b>R-2 (TRAIN)</b>	<b>R-2 (VAL)</b>	<b>MAE (TEST)</b>	<b>MAE (TRAIN)</b>	<b>MAE (VAL)</b>	<b>TRAINING TIME</b>
<b>PHYSICS- BASED MODEL</b>	0.87	-0.285	-	4.67	16.85	-	No training
<b>LINEAR REGRESSION</b>	0.92	0.99	0.92	3.90	1.46	3.90	0.43 s
<b>GRADIENT BOOSTING</b>	0.96	0.99	0.96	2.52	1.53	2.51	1 m 17 s
<b>BAGGING</b>	0.92	0.999	0.92	3.72	0.15	3.71	1 m 32 s
<b>ADA</b>	0.93	0.92	0.93	3.39	3.67	3.39	2 m 23 s
<b>MLPR</b>	0.95	0.93	95	3.15	3.12	3.15	6 s

As can be seen in Table 5-7, all ML models produce better results than the physics-based model on the test set. The physics-based model seems to behave strangely with the train set. Best results are obtained with gradient boosting. Bagging is clearly overfitted, based on the R2 and MAE results. The model output, measured value, and anomaly score of the test set are visualized for the physics-based model, gradient boosting, and linear regression in Figures 5-4, 5-5, and 5-6 respectively. The results of the physics-based model during the training period can also be found in Appendix II.

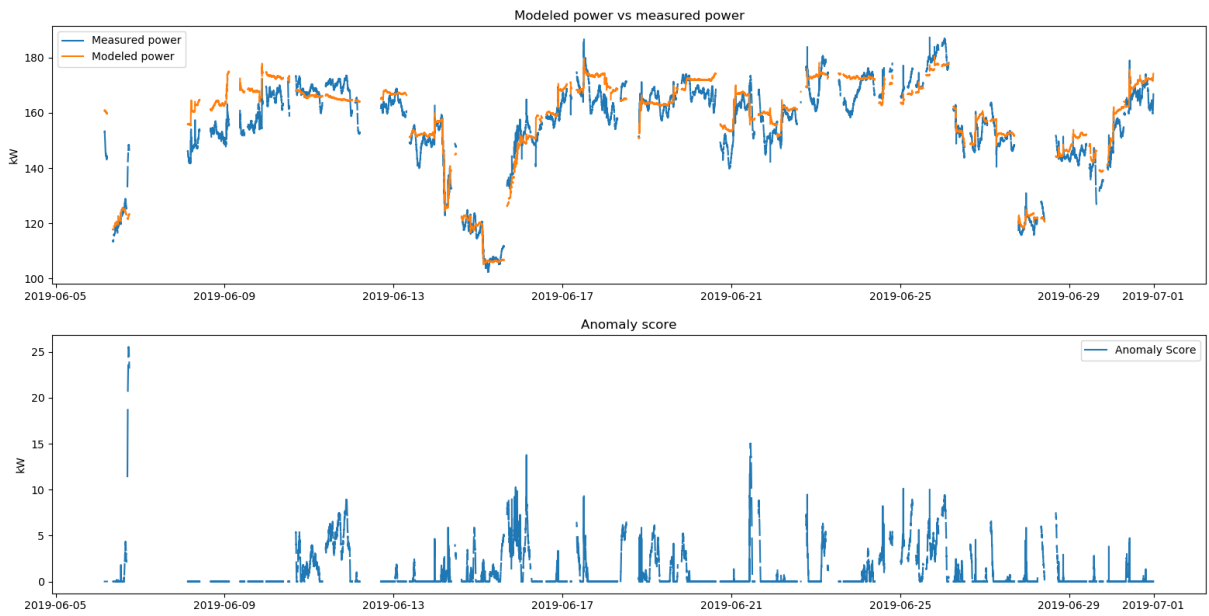


Figure 5-4 Test set for the physics-based model.

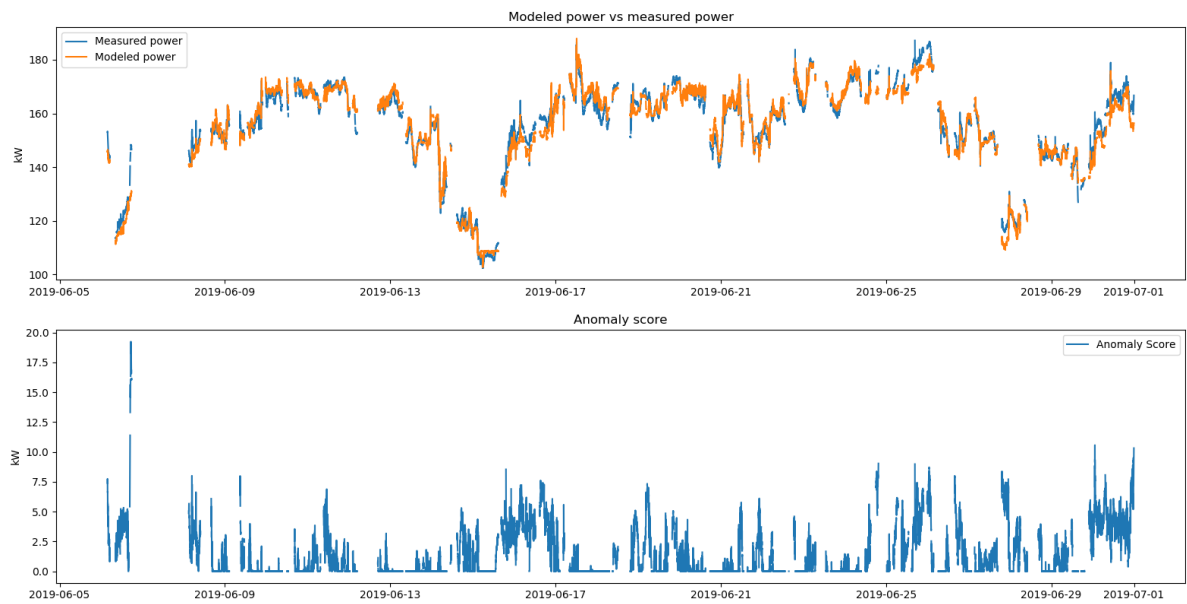
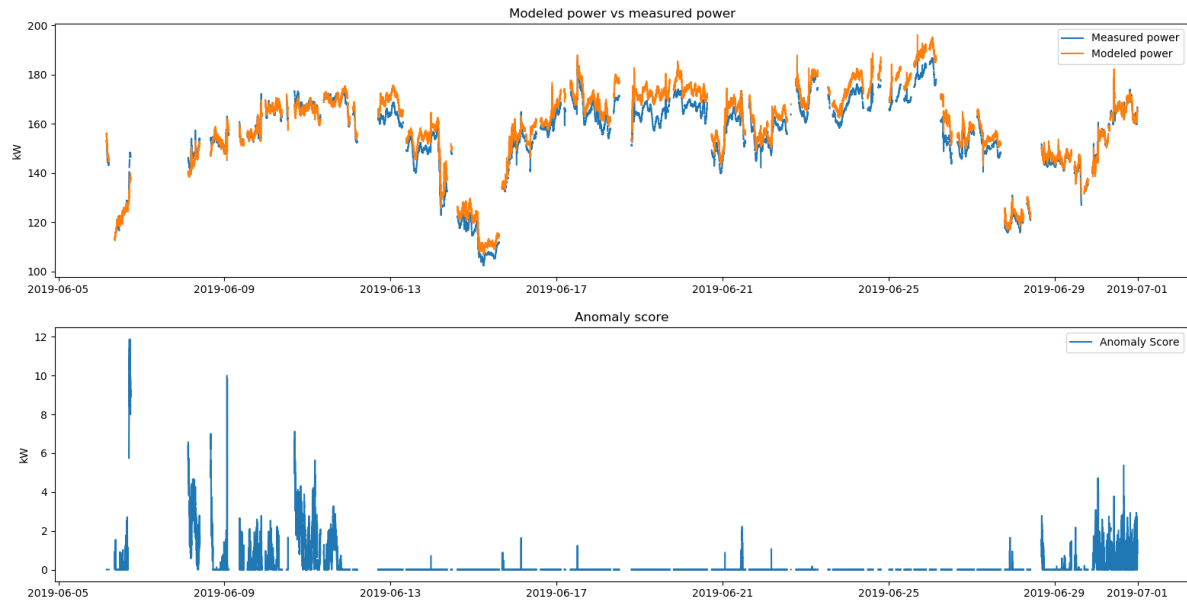


Figure 5-5 Test set for gradient boosting.



*Figure 5-6 Test set for linear regression.*

During the training period, the physics-based model receives a negative R-squared value, because the modeled power is significantly lower than the measured power most of the year. In the testing set, the output of the physics-based model seems decent, and abnormally high electricity consumption can be seen a few times during the test set. The highest anomaly score is found at the start of the test set. During that time, the PM is probably accelerating and shut down, because there are only short periods of continuous data.

Testing periods with gradient boosting and linear regression models can be seen in Figures 5-5 and 5-6 respectively. Anomalies found with gradient boosting are almost the same as those found with the physics-based model. Anomaly scores with the linear regression models are lower than anomaly scores for physics-based model and gradient boosting. The same increase at the start of the test set can also be seen with linear regression.

## 5.5 Hyperparameter tuning

Since there is a lot of data, a relatively small hyperparameter grid is used to minimize training time. The same training and testing sets are used as in the previous chapter. The hyperparameter grid for the grid search can be seen in Table 5-8. The default hyperparameters in Table 5-8 are in bold. Hyperparameters in the grid aim to increase the complexity of the model. The regression problem in hand is complex, because there are many features which have an effect.

Table 5-8 Hyperparameter grid.

HYPERPARAMETER	VALUES
LEARNING_RATE	0.1, 0.05, 0.01
N_ESTIMATORS	100, 500
MAX_DEPTH	3,5,7
MIN_SAMPLES_SPLIT	2,5,10

The hyperparameter tuning results can be seen in Appendix III. The table presented in Appendix III is sorted from smallest to largest by the “mae\_test” column. The best hyperparameters for this task found with the grid search are learning\_rate 0.05, n\_estimators 500, max\_depth 3, and min\_samples\_split 10. The seven best models have 3 as a max\_depth, and the six best models were achieved with 500 n\_estimators. When the learning\_rate is set to 0.01, the models tend to behave poorly. The min\_samples\_split seems to have only a slight effect on model performance. Complex tree structures may require more n\_estimators. Training time grows exponentially as complexity increases. A visualization of the test period with the tuned gradient boosting model can be seen in Figure 5-7.

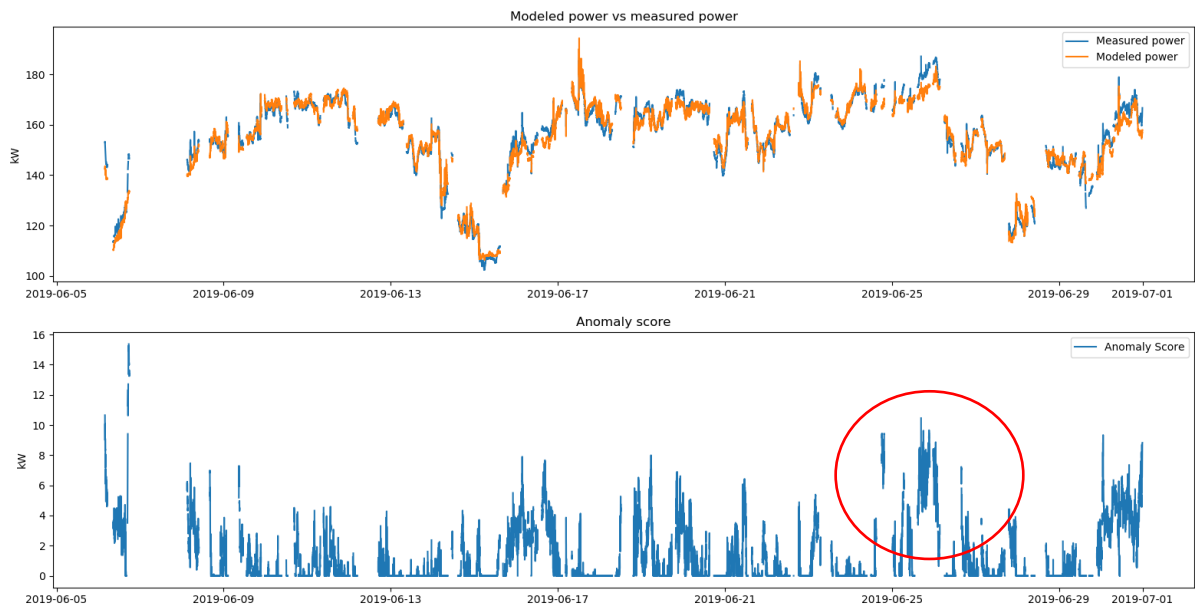


Figure 5-7 Test set of tuned gradient boosting model.

As Figure 5-7 shows, the model behaves in approximately the same way as the model with default hyperparameters in Figure 5-5, except the anomaly score increase after 25.6.2019 (red circle) seems more significant.

## 5.6 Model interpretation with SHAP

The interpretation of the gradient boosting model is done using the SHAP method introduced in Chapter 4.3. The model contains 114 tags, but only a handful make a significant contribution to the model output. The contributions of most of the contributing tags can be seen from the summary plot in Figure 5-8.

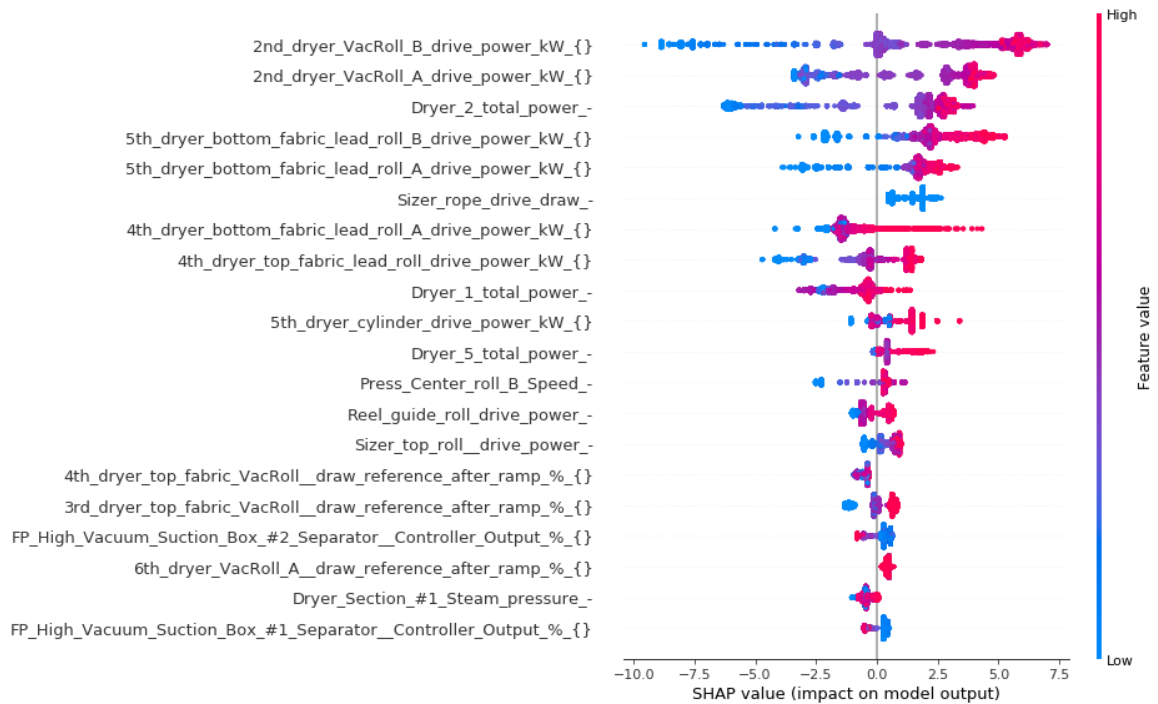


Figure 5-8 SHAP summary plot.

Figure 5-8 shows that the tags contributing most to the prediction are other drives, especially drives from the 2nd dryer group, which is located prior to dryer group 3. Most of the tags have a positive correlation. However, “sizer rope draw”, for example, has a negative correlation. This means that when the draw is increased at the sizer, less power is required from the drives of dryer group 3. The SHAP dependence plot can be seen in Figure 5-9.

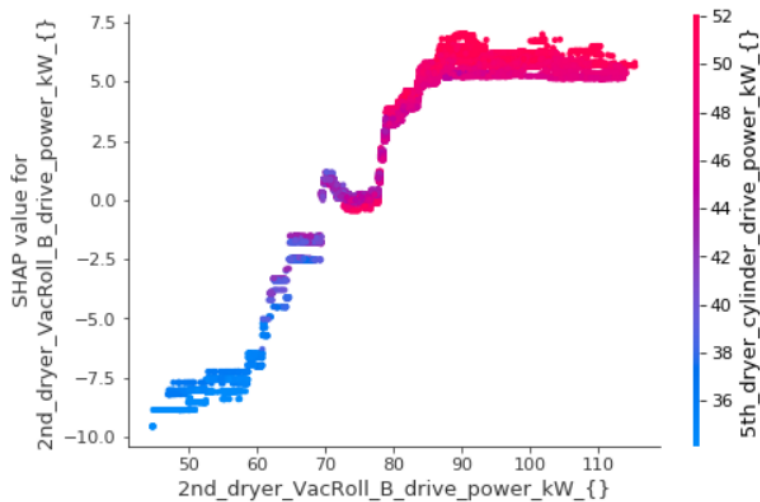


Figure 5-9 SHAP dependence plot.

Figure 5-9 shows that the “2nd\_dryer\_VacRoll\_B\_drive\_power\_kW” contribution to model output is close to zero when it receives values between 70-80. The color of points represents the values of “5th\_dryer\_cylinder\_power\_kW”. Color is used to illustrate how correlated the electricity consumption of these drives is.

## 5.7 Data-driven RCA

In this chapter, a cross-correlation (Chapter 4.4) is used to conduct data-driven RCA. The results for data-driven RCA are described in brief, because there is no information about accidents at this machine. The method is validated in Chapter 6.4, in which the model is built, and data-driven RCA is conducted for the wire section of another PM. Information about failures has been systematically collected from that PM.

In this case, 4,292 different tags are available in the SF database. Calculating cross-correlation, therefore, requires considerable computational power and takes some time. Cross-correlation is calculated during the abnormal period, based on the anomaly score.

The time window for which the cross-correlation is calculated is from -5 to +5 time steps. It translates to the 10-minute window, because the data has a 1-minute resolution. Five minutes is a long time in the papermaking process, and the tags found at the ends of the time window must be assessed carefully. The measurement location at the PM must be considered when assessing results.

Based on the anomaly score in Figure 5-7, cross-correlation is calculated between 25.6.2019 09:00:00 and 26.6.2019 04:00:00. The anomalous period is visualized in Figure 5-10. The period cross-correlation is calculated starting from red and ending with the gray dashed line.

This period is chosen because it starts and ends with an accurate prediction. It may therefore also be possible to see which measures increase/decrease during this period. The 11 and 94 best cross-correlations can be seen in Table 5-9 and Appendix IV respectively, where “CORR” and “T\_SHIFT” represent correlation and time-shift respectively. The correlation for the same tags as in Table 5-9 for all the time shifts can be seen in Appendix V. Only the best correlations with corresponding time shifts are presented in Table 5-9.

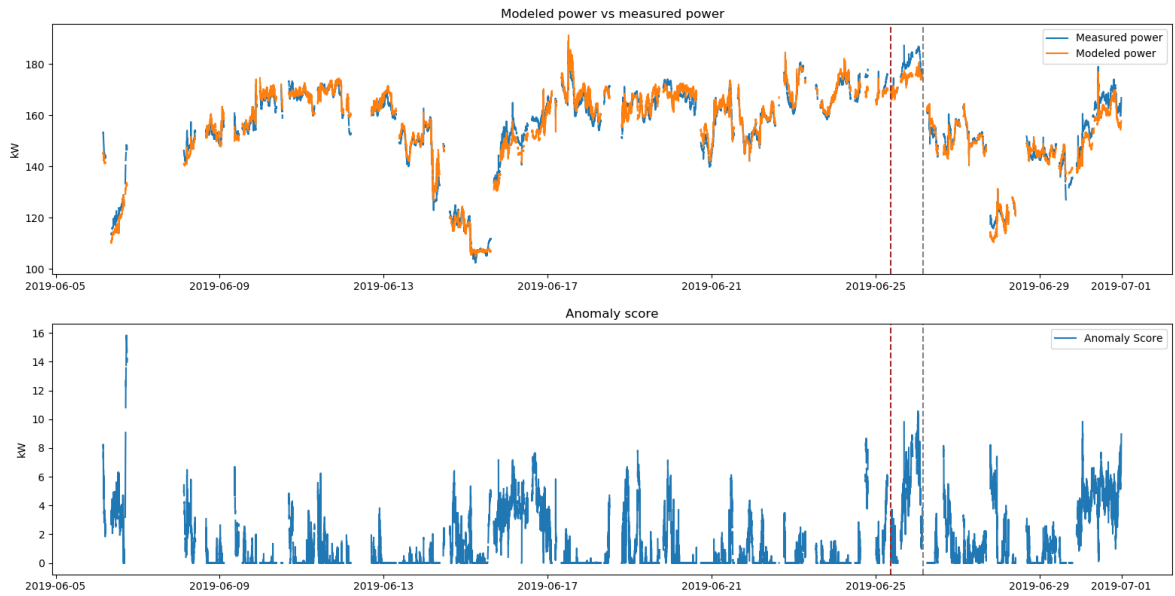


Figure 5-10 Anomalous period found with the gradient boosting method.

Table 5-9 Eleven most cross-correlating tags.

NO.	MEASURE	CORR	T_SHIFT	UNIT
1	4th dryer fabric VacRoll current	0.900	0	
2	4th dryer top fabric VacRoll drive power	0.894	0	kW
3	SYM-Z hydraulics inflow oil temperature control QPV_IN	0.892	5	
4	Tank oil temperature measurement V	0.887	4	
5	Bearing lubrication oil temperature return line measurement V	0.866	-5	
6	4th dryer bottom fabric lead roll A	0.861	0	
7	4th dryer bottom fabric lead roll B	0.861	0	
8	4th dryer bottom fabric lead roll A drive power	0.858	0	kW
9	4th dryer bottom fabric lead roll B drive power	0.858	0	kW
10	Bearing lubrication tank oil temperature measurement V	0.852	5	
11	Bearing lubrication hydraulics inflow oil temperature control QPV_IN	0.842	-5	

The tags corresponding to the drives of dryer group 3 correlate best with the anomaly score. However, they are excluded from Table 5-9, because they are not relevant results. Tags from dryer group 4 also have a high correlation with the anomaly score. Their power also increases during the anomalous period. The other tags found are oil temperature measurements. It makes sense that increased friction in a process may increase lubrication oil temperature. However, the increase in oil temperature is not the root cause. It is possible that the increase in the power of dryer group 3 may be because of deprecated equipment performance. The friction caused by deprecated components can increase the temperature of the lubrication oil. The total power of dryer group 3 and tags 3, 4, 5, 10, and 11 from Table 5-9 are visualized in Figure 5-11. The effects of time shifts are difficult to interpret. Time shifts seem to have little effect on the results anyway when the correlation results of different time shifts in Appendix V are examined.

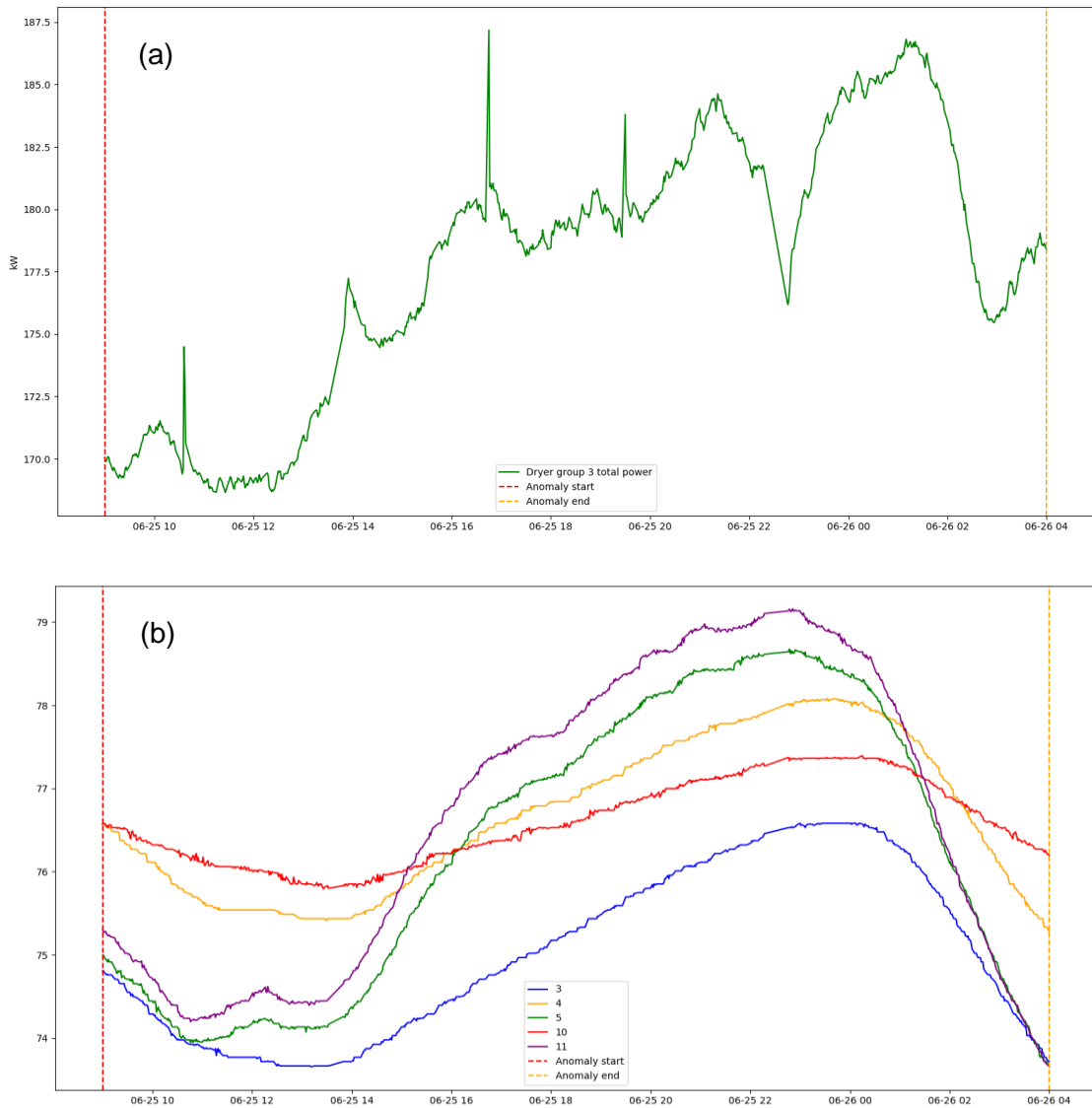


Figure 5-11 Total power of dryer group 3 (a) and correlating tags (b) during anomaly.

## 6 Application

In this chapter, the application done based on proof of concept is described. The application is made to enable continuous monitoring of anomalies and analysis of root causes. Root causes can be analyzed immediately when an anomaly occurs. This application may be a tool for service engineers or a customer product.

The chapter ends with a case study. The model is built for the wire section (see Chapter 2) for the PM that has documented accidents that have occurred at the PM. The target for this model is the total electricity consumption of the wire section drives. In Chapter 5, the target was the total electricity consumption of dryer group 3. In the case study, the application is used to identify the root causes of anomalous periods.

### 6.1 AWS

Trained models are saved in S3 cloud storage, and data is read from SF. Lambda is run every hour. The process inside the Lambda is as follows:

1. Download and process raw data (only tags) from SF (steps described in 5.1 and 5.2)
2. Download model from S3
3. Calculate model output from processed raw data
4. Calculate the anomaly score
5. Save results in SF

The amount of data to be analyzed at each execution consists of only 60 rows, because Lambda is run every hour.

### 6.2 Snowflake data storage

SF works as data storage for raw data, and the results are calculated in Lambda. The desired Tableau dashboard defines the table structure used in SF. The table structure is designed to ensure that the usage of the application is efficient. Two tables are created for SF:

- Model table (Dimension table). Columns: "Model ID", "Mill ID", "Line ID", "Model threshold", and "Section". The table is used to store the individual information of every model. Model ID is the key for combining model information with model results.

- Results table (Fact table). Columns: “Timestamp”, “Model ID”, “Model output”, and “Measured value”. This table contains the results for all the models.

The table structure enables the addition of new models and the efficient visualization of results.

### 6.3 Tableau dashboard

Tableau visualization is designed by following common VII guidelines. An intuitive tableau makes interpreting results easy and efficient. Tableau visualization can be seen in Figure 6-1. All sensitive information has been filtered out from Figure 6-1.



Figure 6-1 Tableau dashboard

In Figure 6-1, there are filtering options at the top section of the view, where the user can select mill id, line id, a section from the machine, and the time period. Plots “model vs. measurement” and “anomaly score” are rendered based on these filters. The third plot, “selected\_tag” is rendered based on user selection from the “correlating tags” view from the left-hand side of the dashboard.

The yellow background represents the time period for which the correlation is calculated. The user can calculate correlation for any period he/she desires. Filters to control the correlation calculation can be found above the “correlating tags” view. The user can visualize tags

he/she wants simply by clicking the row from the “correlating tags” view. Tags are sorted in descending order by the absolute correlation. SMEs can easily scroll through correlating tags and study their relevance for the anomaly.

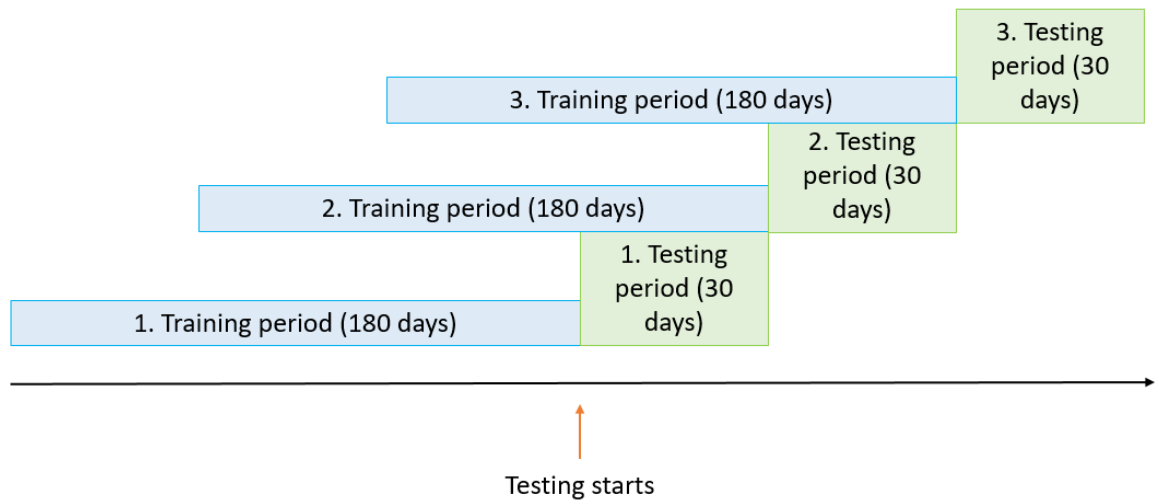
## 6.4 Case study

In this chapter, the previously introduced methods are used to discern if abnormal behavior is present before failures. The application is used to see if data-driven RCA helps to identify root causes before failures. The model is now built for the wire section (Chapter 2), because the necessary drive tags to build a model were found only for the wire section on this PM.

This time processing is a little easier, because this PM's data is almost complete. Only one period is missing between early April and early May, so data imputation is not required. Furthermore, tags that are included in the model are located at the wire section, and the best model was acquired with all the tags selected by the SMEs. The hyperparameter grid is optimized based on the results in Chapter 5.5. This time some of the default hyperparameters are excluded from the grid search. However, one model is trained with default hyperparameters as a benchmark. Only a slight improvement from the default model was achieved. The best hyperparameters are: `learning_rate` 0.02; `n_estimators` 500; `max_depth` 3; and `min_samples_split` 10. The `min_samples_split` was set to 10, because it had no significant influence in Chapter 5.5. The hyperparameter grid can be found in Appendix VI. Grid search results can be found in Appendix VII.

The SHAP summary plot in Appendix VIII shows that speed is a dominant feature. The model's decision making is based mostly on speed. The dependence plot for speed can be seen in Appendix IX. When the speed decreases, SHAP values also decrease. There seems to be almost a linear correlation. Speed values are colored based on basis weight. The coloring illustrates that at lower basis weights, the machine runs at higher speeds.

To simulate the actual usage of this concept, the model is always trained, using the 180 previous days. The trained model is then used for the next 30 days, and the model is trained again after 30 days have elapsed. This is because the operators are constantly tuning the machine settings to improve productivity. A visualization of the actual usage of this method can be seen in Figure 6-2.



*Figure 6-2 Training and testing periods.*

Testing starts from 1.1.2019 and ends on 1.7.2019. The major failures at the wire section and maintenance actions performed at the PM during this period are presented in Tables 6-1 and 6-2 respectively. It is important to be aware that maintenance actions take place all over the PM, not just at the wire section. The model output, measured value, anomaly score, a 7-day moving average of anomaly score, major failures, and maintenance actions can be seen in Figure 6-3.

*Table 6-1 Failures at the wire section.*

<b>NO</b>	<b>FAILURE</b>	<b>START</b>	<b>END</b>	<b>DURATION (MIN)</b>
<b>1</b>	Electrical Defect Wire Part	05.01.19 6:45	05.01.19 7:43	57
<b>2</b>	Mechanical Defect Wire Part	05.01.19 12:28	05.01.19 12:52	24
<b>3</b>	Electrical Defect Wire Part	23.03.19 20:37	24.03.19 14:38	1081
<b>4</b>	Electrical Defect Wire Part	27.05.19 7:09	27.05.19 8:54	105
<b>5</b>	Electrical Defect Wire Part	27.06.19 1:08	27.06.19 5:07	239

Table 6-2 Maintenance action at the PM.

NO	MAINTENANCE ACTION	START	END	DURATION (MIN)
1	Planned maintenance shutdown	22.01.19 5:47	22.01.19 16:59	672
2	Unplanned maintenance	22.01.19 17:00	22.01.19 17:59	60
3	Unplanned maintenance	22.01.19 18:00	22.01.19 18:59	60
4	Unplanned maintenance	22.01.19 19:00	22.01.19 21:42	163
5	Unplanned change of clothing at wire	23.01.19 8:41	23.01.19 11:53	192
6	Planned maintenance shutdown	26.02.19 5:39	26.02.19 16:59	680
7	Unplanned maintenance	26.02.19 17:00	26.02.19 20:58	239
8	Planned maintenance shutdown	02.04.19 5:42	02.04.19 18:59	798
9	Unplanned maintenance	02.04.19 19:00	02.04.19 19:48	49
10	Unplanned change of clothing at wire	03.04.19 3:07	03.04.19 8:22	315
11	Planned maintenance shutdown	16.05.19 9:00	16.05.19 12:59	239
12	Unplanned maintenance	16.05.19 13:00	16.05.19 14:26	87
13	Planned maintenance shutdown	18.06.19 5:41	18.06.19 21:00	919
14	Unplanned maintenance	18.06.19 21:01	19.06.19 5:08	487



Figure 6-3 Test period results for the wire section model.

Figure 6-3 shows that the anomaly score starts to increase more than two weeks before failures 3 and 4 (Table 6-1) occur. It is also noteworthy that the anomaly score starts to decrease when maintenance actions are conducted after failures 3 and 4. There is a slight increase in the anomaly score before failures 1 and 2, but it is not significant enough for it to stand out. Failure 5 occurs when the anomaly score is close to 0. Failures can be quite different, and some of the failures cannot be seen in electricity consumption.

The next step is to calculate the correlation between the anomaly score and all the tags. The correlation is calculated for periods where possible problems can be captured. These periods are referred to as “study periods”. Study periods are selected so that they consist of an increase in the anomaly score. Tags that are increasing/decreasing simultaneously are thus captured. Study periods are listed in Table 6-3 and visualized in Figure 6-4. Time shifts are not used when calculating correlation, because the data has a 10-minute resolution.

Table 6-3 Study periods.

STUDY PERIOD	START	END
1	2019-02-24 00:00:00	2019-02-24 06:00:00
2	2019-02-28 00:00:00	2019-02-28 18:00:00
3	2019-03-02 00:00:00	2019-03-04 00:00:00
4	2019-03-08 00:00:00	2019-03-12 12:00:00
5	2019-03-19 02:00:00	2019-03-19 18:00:00
6	2019-03-29 00:00:00	2019-03-30 00:00:00
7	2019-05-11 06:00:00	2019-05-12 03:00:00
8	2019-05-19 00:00:00	2019-05-20 03:00:00

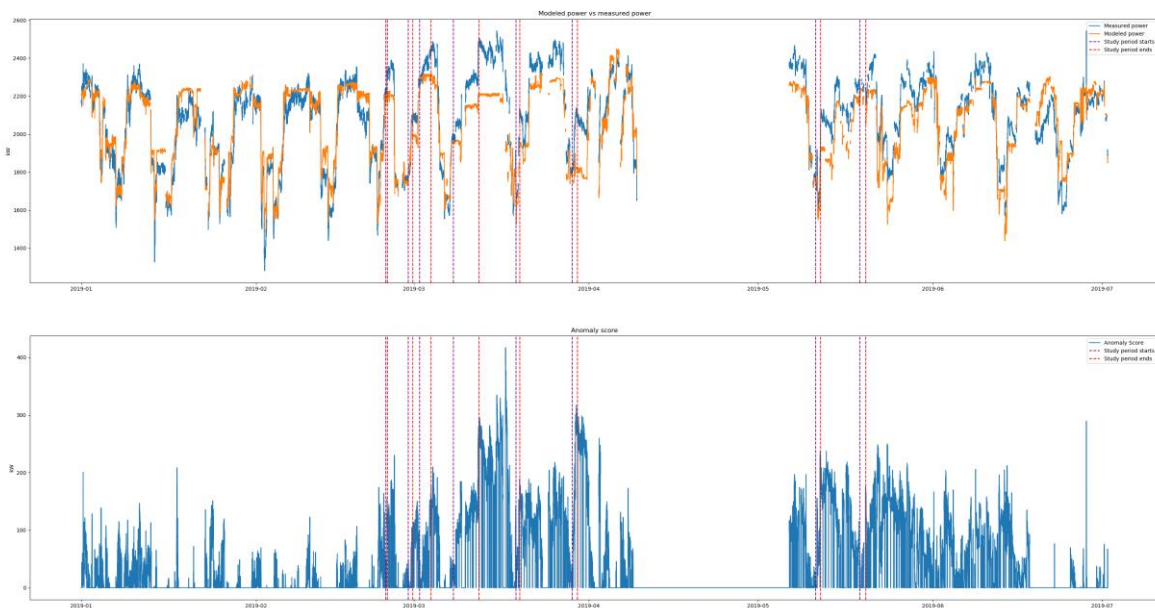


Figure 6-4 Study periods.

The twenty most correlating tags of each study period are presented in Appendix X. The study periods were examined by the SMEs with the application created during the thesis. Unfortunately, the root causes of the failures could not be identified due to a lack of knowledge about the failures. However, the most probable tags for power increase in study period 3 were identified. Model output, measured power, and three tags selected by SMEs are visualized in Figure 6-5. Tags a, b, and c were selected as the most likely causes for the increase in power during study period 3. Tags a, b, and c are not included in the twenty most

correlating tags of study period 3 in Appendix X. The SMEs found these tags with the application by reviewing the “correlating tags” view (Figure 6-1).



Figure 6-5 Study period 3 results.

In Figure 6-5, study period 3 is the period where the graphs have a yellow background. During the period, measured power starts below the modeled power and increases quite steadily throughout the period. The tags in Figure 6-5 are:

- a, reject flow from machine screen
- b, amount of cellulose in long fiber mass
- c, top layer mixing tank mass consistency

A possible scenario the SMEs came up with is that the mass arriving at the wire section changes, meaning the properties of the paper web change. Changes in the paper web affect the friction between fabrics and vacuum units in the wire section. Increased friction between vacuum units and fabrics increases the drives' power demand. Friction between fabrics and vacuum units cannot be measured, so it is not included in the model. The SMEs' analysis of study period 3 made with the tool illustrates how the root causes of anomalies can be identified.

## 7 Conclusions

Most of the PMs are different, and data collection is undertaken in many ways. Input data should be standardized to facilitate the adding of new PMs to the tool. In Chapter 5.2, multivariate data imputation, MICE, improved model accuracy. It is still questionable if imputed data can be used in customer applications, because they are being used in decision making. Many VII applications could still use imputed data. In future, data imputation could be implemented in the VII data pipeline as an option.

Feature selection improved model accuracy. However, at Valmet, the role of SMEs cannot be disregarded in feature selection. In many cases, feature selection methods include tags that cannot be used in the model for various reasons. The selection of tags has to be done with the SME, but the tags selected by the SME can be reduced using feature selection methods. Low variance and RReliefF led to the best performance of the feature selection methods used in the empirical part of this thesis.

The gradient boosting model had the best performance of all models. The accuracy of the physics-based model was poorer than the ML models. Yet the first approach for anomaly detection introduced in Chapter 4 is more reliable with the physics-based model than with ML models, because the effect of each tag in the physics-based model is carefully determined. The ML algorithm does not necessarily learn the effect of each individual tag correctly, despite its good performance. The hyperparameter tuning of the gradient boosting model did not improve model accuracy significantly. A more thorough grid search or other hyperparameter search methods could be applied in the future. A Bayesian optimization or random search could speed up the hyperparameter optimization.

The SHAP method, used in Chapter 5.6, can be used to evaluate each tag's effect on the ML model output. Regarding SHAP, the largest contributor to decision making of the dryer group 3 model was the vacuum roll drive from dryer group 2. The high correlation between the drives of the adjacent groups produced an accurate model. However, its capacity to detect anomalies within the dryer group 3 is questionable. The most correlating tags found for dryer group 3 were credible root causes. Increased oil temperatures seem likely, and it may be due to degraded bearings, but it may be just a coincidence. Unfortunately, there is no way to discover what happened at the PM during that anomalous period. Cross-correlation may be useful if the user has precise knowledge of the PM. However, in this thesis, correlation without timeshifts produces almost identical results to cross-correlation. Cross-correlation might be more valuable for data with a higher frequency.

The tags found in the case study (Chapter 6.4) may explain the anomaly during study period 3. SMEs used the application to identify anomalous tags, but the identified tags were not the most correlating. The root causes of the failures at the wire section could not be identified, due to a lack of information about the failures. It is rare to find tags that clearly indicate that a component is going to fail.

It was noticed that abnormally behaving tags do not necessarily correlate with the anomaly score. In future, other methods could, therefore, be added to data-driven RCA. For example, variance, average, minimum, and maximum values could be calculated for every tag during a normal and anomalous period. The relative difference between tag values in a normal and anomalous period might reveal the root causes of anomalies.

The threshold for early warnings of failures could also be implemented. Zhao et al. (2019) applied extreme value theory in designing the early warning threshold. A similar approach might be useful in the application created during this thesis.

Feature extraction was left out from this thesis. However, extracted features could produce better models than plain tag data. In future, some features could be calculated in the data pipeline.

## 8 Summary

The goal of this project was to develop methods to identify deprecated components or problem areas in the process. This was achieved by modeling the electricity consumption of drives and calculating the correlation between the anomaly score and tags.

The first step was to become familiar with electric drives (Chapter 2) and what affected their electricity consumption. The VII was introduced (Chapter 3), because it was utilized in both data acquisition and the creation of the application. Chapter 4 contained the literature review. It contained four main sections: data preprocessing; ML models; model interpretation; and root cause analysis.

In Chapter 5, PoC was created, based on the theoretical framework. Tags affecting electricity consumption were limited by the SMEs. The selected tags were reduced using feature selection methods. Various ML models were built and compared, along with the physics-based physics-based model. Gradient boosting was selected as the best method, based on its accuracy with unseen data. The hyperparameters of the gradient boosting model were optimized using a grid search. The model was then interpreted using the SHAP method, which enables the interpretation of complex ML models.

Anomalous periods can be found by comparing the model output to the measured value. Data-driven RCA was used to discover the root causes of anomalies. Data-driven RCA was performed by calculating the correlation during anomalous periods between the anomaly score and all the tag values collected from the PM. The anomaly score is the difference between model output and measured value. The root causes of anomalies may be identified from correlating tags. Cross-correlation (Chapter 4.4) may also be utilized to capture root causes.

Finally, an application built during the thesis was introduced in Chapter 6. The application uses AWS back-end, SF data storage, and Tableau dashboard to present the results. Correlation between an anomaly score and all the tags from a PM can be calculated in the Tableau dashboard for a user-specified period. The application is created for SMEs who know the papermaking process and are capable of making conclusions based on tag data.

The application was used in the case study (Chapter 6.4). In the case study, the methods introduced in Chapter 5 were applied to another PM's wire section. The model was built, and the anomalies were studied before known failures. The anomaly score increased before two of the failures. SMEs used the application to identify the possible root causes of the anomalies. SMEs came up with a possible scenario for one of the anomalies, but it did not

explain the failures. The created scenario illustrates how the application can be used to identify the root causes of the data. Although the root causes were not identified, this does not mean the correlation should not be used in data-driven RCA.

For future research, methods other than correlation could be experimented with in data-driven RCA. Also, feature extraction could be included in the data pipeline to make developers' life easier. The early warning threshold for anomalies could also be implemented in the application.

## References

Alasdair G. 2016. Industry 4.0. Apress

Bach, S., Binder, A., Montavon, G., Klauschen, F., Klaus-Robert Müller & Samek, W. 2015, On Pixel-Wise Explanations for Non-Linear Classifier Decisions by Layer-Wise Relevance Propagation, *PLoS One*, vol. 10, no. 7.

Boker, S.M., Xu, M., Rotondo, J.L & King, K. 2002. Windowed cross-correlation and peak picking for the analysis of variability in the association between behavioral time series. *Psychological methods*. vol. 7. issue 3. pp 338 - 55.

Breiman, L. 1996. Bagging predictors. *Machine Learning*. vol. 24, pp. 123-140

Datta, A. Sen, S. & Zick Y. 2016. Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems. *IEEE Symposium on Security and Privacy*. pp. 598-617

Efron, B. & Tibshirani, R.J. 1994. *An introduction to the Bootstrap*. CRC Press.

Ghiselli, E. E. 1964. *Theory of Psychological Measurement*. McGraw-Hill

Godin, F., Degrave, J., Dambre, J. & De Neve, W. 2017. Dual Rectified Linear Units (DReLU): A Replacement for Tanh Activation Functions in Quasi-Recurrent Neural Networks. *Pattern Recognition Letters*. vol. 116, pp. 8-14

Guyon, I., Weston, J., Barnhill, S. & Vapnik, V. 2002. Gene Selection for Cancer Classification using Support Vector Machines. vol. 46, issue 1-3, pp. 389 – 422

Gracia, S., Luengo, J. & Herrera, F. 2015. *Data Preprocessing in Data Mining*. Springer, Cham

Guzel, M., Kok, I., Akay, D. & Ozdemir, S. 2019. ANFIS and Deep Learning-based missing sensor data prediction in IoT. *Concurrency Computat: Practice and Experience*.

Hall, M.A. & Smith, L.A. 1999. Feature selection for machine learning: comparing a correlation-based filter approach to the wrapper. In *FLAIRS*. pp. 235–239.

Hastie, T., Tibshirani, R. & Friedman, J. 2009. *The Elements of Statistical Learning*. Springer. New York

Hayes, B. 2019. Programming Languages Most Used and Recommended by Data Scientists. [online document]. [Accessed 7.11.2019]. Available at: <https://businessoverbroadway.com/2019/01/13/programming-languages-most-used-and-recommended-by-data-scientists/>

Karjalainen, P. 1998. Paperikoneen sähkökäyttöjen mitoitus. Master's Thesis. University of Oulu

Krawczak, M. 2013. Multilayer Neural Networks. Springer

Kumar A., Shankar R. & Thakur L. S. 2018. A big data-driven sustainable manufacturing framework for condition-based maintenance prediction. *Journal of Computational Science*, vol. 27, pp. 428-439.

Lei, Y., Li, N., Guo, L., Li, N., Yan, T. & Lin, J. 2018. Machinery health prognostics: A systematic review from data acquisition to RUL prediction. *Mechanical Systems and Signal Processing*, vol. 104, pp. 799-834.

Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino R.P., Tang, J. & Liu, H. 2016. Feature Selection: A Data Perspective. *ACM Computing Surveys*. vol. 50. Issue 6.

Lipovetsky, S. & Conklin, M. 2001. Analysis of regression in game theory approach. *Applied Stochastic Models in Business and Industry*. vol. 17. Issue 4. pp. 319-330

Lundberg, S. & Lee, S. 2017. A Unified Approach to Interpreting Model Predictions. *Advances in Neural Information Processing Systems*. pp. 4766-4775

Matplotlib. 2019. Frontpage. [online document] [Accessed 13.11.2019] Available at <https://matplotlib.org/>

Mobley, R.K. 1999. Root Cause Failure Analysis. Butterworth-Heinemann

Mobley, R. K. 2002. An Introduction to Predictive Maintenance. 2<sup>nd</sup> ed. Butterworth-Heinemann

Mehrotra, K.G., Mohan, C.K. & Huang, H. 2017. Anomaly Detection Principles and Algorithms. Springer

Nielsen, M. 2015. Neural networks and deep learning. Determination press

Rumelhart, D.E, Hintor G.E. & Williams R.J. 1986. Learning representations by back-propagating errors. *Nature*, vol. 323, pp 533-536

Numpy. 2019. Frontpage. [online document] [Accessed 13.11.2019] Available at <https://numpy.org/>

Okes, D. 2009. Root Cause Analysis - The Core of Problem Solving and Corrective. ASQ Quality Press

Pandas. 2019. Frontpage. [online document] [Accessed 13.11.2019] Available at <https://pandas.pydata.org/>

Rathi A. 2018. Dealing with Noisy Data in Data Science. [online document]. [Accessed 7.11.2019]. Available at <https://medium.com/analytics-vidhya/dealing-with-noisy-data-in-data-science-e177a4e32621>

Rebala, G., Ravi, A. & Churiwala, S. 2019. An introduction to Machine Learning. Springer, Cham

Ribeiro, M. T., Singh S. & Guestrin C. 2016. "Why should I trust you?" Explaining the predictions of any classifier. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.

Robnik-Šikonjalgor, M. & Kononenko, I. 2003. Theoretical and Empirical Analysis of ReliefF and RReliefF. Machine Learning, vol. 53, pp. 23-69

Ryan, T.P. 2009. Modern Regression Methods. 2<sup>nd</sup> ed. John Wiley & Sons.

Sáez, J.A., Galar, M., Luengo, J. & Herrera, F. 2013. Tackling the problem of classification with noisy data using Multiple Classifier Systems: Analysis of the performance and robustness. Information Sciences, vol. 247, pp 1-20

Scikit-learn. 2019. Frontpage. [online document]. [Accessed 7.11.2019] Available at <https://scikit-learn.org/stable/>

Shapley, L.S. 1952. A value for n-person games. RAND Corporation

Shrikumar, A., Greenside, P. & Kundaje, A. 2017. Learning important features through propagating activation differences. 34th International Conference on Machine Learning. vol 7. pp. 4844-4866

Singh, H. 2018. Understanding Gradient Boosting Machines. [online document]. [Accessed 7.11.2019] Available at <https://towardsdatascience.com/understanding-gradient-boosting-machines-9be756fe76ab>

Spyder. 2019. Front page. [online document]. [Accessed 13.11.2019] Available at <https://www.spyder-ide.org/>

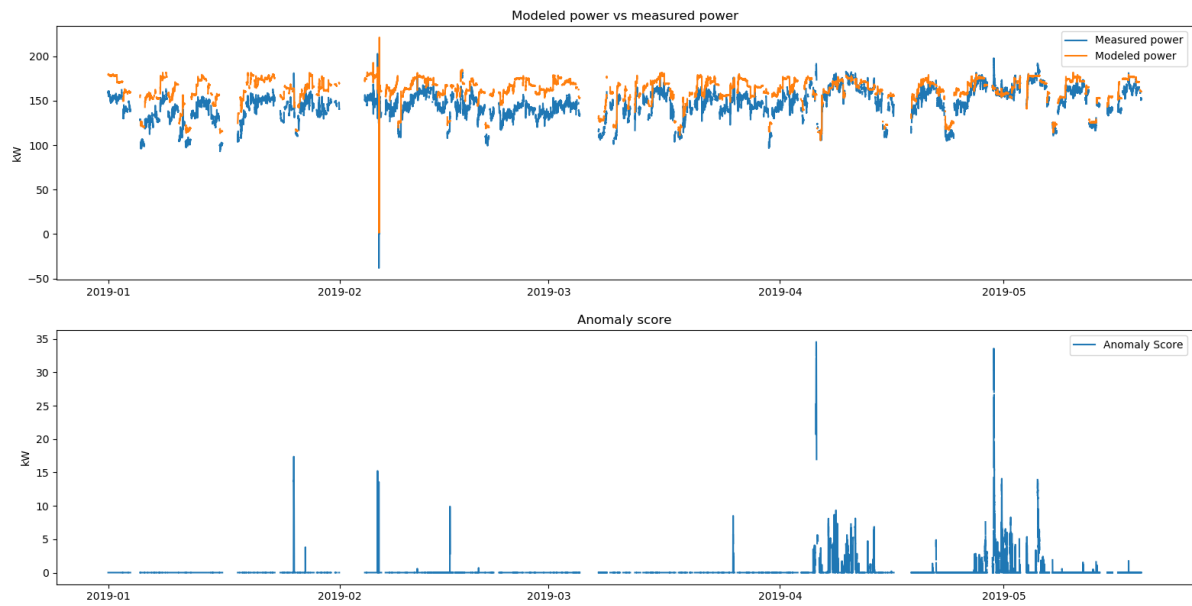
Steiner, S., Zeng, Y., Young, T.M., Edwards, D.J., Guess, F.M. & Chen, C. 2017. A Study of Missing Data Imputation in Predictive Modeling of a Wood-Composite Manufacturing Process. Journal of Quality Technology. vol. 48. Issue 3. pp. 284-296

- Strumbelj, E. & Kononenko, I. 2014. Explaining prediction models and individual predictions with feature contributions. *Knowledge and information systems*. vol. 41. pp. 647-665
- Tibshirani, R. 1996. Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society*. vol. 58, No. 1, pp. 267 – 288
- Tutz, G. & Ramzan, S. 2015. Improved methods for the imputation of missing data by nearest neighbor methods. *Computational Statistics & Data Analysis*. vol. 90 pp. 84-99
- Van Buuren, S. & Groothuis-Oudshoorn, K. 2011. mice: Multivariate Imputation by Chained Equations in R. *Journal of Statistical Software*. vol. 45, pp. 1–67
- Valmet internal sources. 2019. Interviews, e-mail correspondence & expert reviews.
- Valmet 2018. Valmet's Annual Review, 2018. Espoo, Valmet Oyj
- Zhao, H., Liu, H., Hu, W. & Yan, X. 2018. Anomaly detection and fault analysis of wind turbine components based on deep learning network. *Renewable Energy*. vol 127, pp. 825-834
- Zhu, X. & Wu, X. 2004. *Class Noise vs. Attribute Noise: A Quantitative Study*, *Artificial Intelligence Review*, vol. 22, pp. 177-210

## Appendix I – SQL query

```
SELECT date_trunc('hour',
"VALMET_EDW_PROD"."PUBLIC"."F_PM_AND_BM_TAG_VALUE".TAG_TIMESTAMP) AS
HOUR_TIMESTAMP,
((extract(minute FROM
"VALMET_EDW_PROD"."PUBLIC"."F_PM_AND_BM_TAG_VALUE".TAG_TIMESTAMP)::int
/ 1))::int AS RESO_SLOT,
dateadd(minute, RESO_SLOT * 1, hour_timestamp) AS TIMESTAMP,
AVG("VALMET_EDW_PROD"."PUBLIC"."F_PM_AND_BM_TAG_VALUE".TAG_VALUE) AS
TAG_VALUE,
"VALMET_EDW_PROD"."PUBLIC"."D_TAG".CUSTOMER_TAG AS TAG_NAME
FROM "VALMET_EDW_PROD"."PUBLIC"."F_PM_AND_BM_TAG_VALUE"
LEFT OUTER JOIN "VALMET_EDW_PROD"."PUBLIC"."D_TAG"
ON "VALMET_EDW_PROD"."PUBLIC"."F_PM_AND_BM_TAG_VALUE".TAG_KEY =
"VALMET_EDW_PROD"."PUBLIC"."D_TAG".TAG_KEY
WHERE millid = '123123' and TAG_NAME in ('tag1', tag2', ....., 'tagn')
AND
TIMESTAMP between '2018-09-14' AND '2019-09-15'
GROUP BY HOUR_TIMESTAMP, RESO_SLOT, TIMESTAMP, TAG_NAME
ORDER BY TIMESTAMP, RESO_SLOT ASC;
```

## Appendix II – Train set for physics-based model



## Appendix III – Grid search results

n_estimators	learning_rate	max_depth	min_samples_split	mae_test	mae_train	mae_val	r2_test	r2_train	r2_val
500	0.05	3	10	2.258	1.186	1.213	0.969	0.991	0.990
500	0.05	3	5	2.315	1.187	1.213	0.968	0.991	0.990
500	0.05	3	2	2.328	1.184	1.211	0.967	0.991	0.989
500	0.1	3	2	2.400	0.972	1.010	0.966	0.994	0.993
500	0.1	3	10	2.434	0.972	1.011	0.966	0.994	0.993
500	0.1	3	5	2.497	0.970	1.010	0.963	0.994	0.991
100	0.1	3	5	2.707	1.580	1.604	0.958	0.985	0.982
500	0.1	5	2	2.710	0.589	0.656	0.960	0.998	0.993
100	0.1	3	10	2.735	1.580	1.605	0.957	0.985	0.982
100	0.05	5	5	2.752	1.230	1.266	0.957	0.991	0.987
100	0.05	5	10	2.764	1.230	1.267	0.956	0.991	0.987
500	0.05	7	10	2.775	0.465	0.544	0.955	0.999	0.997
500	0.01	5	10	2.788	1.217	1.252	0.957	0.991	0.987
100	0.05	7	2	2.807	0.826	0.869	0.954	0.996	0.994
500	0.01	5	5	2.808	1.216	1.252	0.956	0.991	0.987
500	0.05	7	2	2.817	0.469	0.547	0.951	0.999	0.996
100	0.05	5	2	2.821	1.229	1.264	0.955	0.991	0.987
500	0.05	5	5	2.830	0.747	0.799	0.955	0.997	0.994
500	0.01	5	2	2.833	1.215	1.250	0.955	0.991	0.987
100	0.05	7	10	2.837	0.829	0.874	0.954	0.996	0.993
100	0.1	5	2	2.839	1.015	1.056	0.954	0.994	0.989
100	0.1	3	2	2.849	1.580	1.606	0.951	0.985	0.982
500	0.05	5	2	2.854	0.746	0.798	0.955	0.997	0.993
500	0.05	7	5	2.881	0.469	0.548	0.949	0.999	0.996
500	0.05	5	10	2.884	0.745	0.796	0.955	0.997	0.993
500	0.1	5	10	2.943	0.586	0.649	0.953	0.998	0.996
500	0.1	5	5	2.951	0.586	0.650	0.952	0.998	0.995
100	0.1	5	10	2.953	1.028	1.065	0.952	0.994	0.991
100	0.05	7	5	2.977	0.829	0.874	0.947	0.996	0.993
500	0.01	3	2	2.985	1.943	1.953	0.948	0.977	0.973
500	0.01	3	5	2.988	1.943	1.953	0.948	0.977	0.973
500	0.01	3	10	2.993	1.943	1.954	0.948	0.977	0.973
100	0.1	5	5	3.032	1.025	1.062	0.949	0.994	0.991
500	0.01	7	5	3.098	0.820	0.864	0.940	0.996	0.993
500	0.01	7	2	3.152	0.822	0.867	0.938	0.996	0.993
500	0.01	7	10	3.152	0.822	0.867	0.938	0.996	0.993
500	0.1	7	10	3.259	0.361	0.472	0.931	0.999	0.997
100	0.05	3	5	3.331	1.949	1.965	0.937	0.977	0.974
100	0.05	3	2	3.331	1.949	1.965	0.937	0.977	0.974
100	0.05	3	10	3.331	1.949	1.966	0.937	0.977	0.973
100	0.1	7	2	3.432	0.666	0.718	0.927	0.997	0.995

500	0.1	7	2	3.470	0.360	0.478	0.924	0.999	0.995
100	0.1	7	5	3.554	0.658	0.709	0.919	0.997	0.994
100	0.1	7	10	3.557	0.666	0.718	0.917	0.997	0.995
500	0.1	7	5	3.559	0.359	0.473	0.917	0.999	0.997
100	0.01	5	10	6.218	5.325	5.247	0.787	0.827	0.822
100	0.01	5	5	6.224	5.325	5.247	0.786	0.827	0.822
100	0.01	5	2	6.227	5.325	5.247	0.786	0.827	0.822
100	0.01	3	5	6.636	6.007	5.911	0.754	0.773	0.769
100	0.01	3	10	6.636	6.007	5.911	0.754	0.773	0.769
100	0.01	3	2	6.636	6.007	5.910	0.754	0.773	0.769
100	0.01	7	10	7.094	4.976	4.902	0.732	0.847	0.844
100	0.01	7	2	7.098	4.976	4.902	0.730	0.847	0.844
100	0.01	7	5	7.123	4.976	4.901	0.730	0.847	0.844

## Appendix IV – Cross-correlation results

Description	Corr	T_SHIFT	Unit
3rd dryer top fabric lead roll drive power	0.900	0	kW {}
3rd dryer top fabric VacRoll drive power	0.894	0	kW {}
Dryer 3 total power	0.892	0	
3rd dryer top fabric VacRoll current	0.887	0	
3rd dryer top fabric lead roll	0.866	0	
3rd dryer bottom fabric lead roll A drive power	0.861	0	kW {}
4th dryer fabric VacRoll current	0.861	0	
3rd dryer bottom fabric lead roll A	0.858	0	
3rd dryer bottom fabric lead roll B	0.858	0	
4th dryer top fabric VacRoll drive power	0.852	0	kW {}
SYM-Z hydraulics inflow oil temperature control QPV_IN	0.842	5	
Tank oil temperature measurement V	0.834	4	
Bearing lubrication oil temperature return line measurement V	0.814	-5	
4th dryer bottom fabric lead roll A	0.813	0	
4th dryer bottom fabric lead roll B	0.812	0	
4th dryer bottom fabric lead roll A drive power	0.810	0	kW {}
4th dryer bottom fabric lead roll B drive power	0.810	0	kW {}
Bearing lubrication tank oil temperature measurement V	0.809	5	
Bearing lubrication hydraulics inflow oil temperature control QPV_IN	0.805	-5	
1st press loading control DS LMN	0.798	-1	% {}
1st press loading control TS LMN	0.797	-3	% {}
5th dryer cylinder drive power	0.794	5	kW {}
Actuator average temperature	0.794	-3	
Sheet break detection 7th dryer group cylinder 48 DS	0.785	-5	
Finish ply headbox slice beam heating U	0.780	5	
Finish ply headbox slice beam heating PV_IN	0.778	5	
Roll cooling return oil temperature V	0.778	-5	
Hot oil system heating and cooling control SP	0.775	5	
SS starch storage tank temperature U	0.774	-5	
Hot oil system heating and cooling control QPV_IN	0.774	5	
Hot oil system feed flow oil temperature V	0.773	5	
2nd press loading control TS LMN	0.762	-5	% {}
Dry end lubrication temperature LMN	0.757	1	
37th dryer cylinder doctor loading U	0.756	4	
37th dryer cylinder doctor loading U	0.751	4	PSI {}
5th dryer cylinder	0.750	5	
Sheet break detection 7th dryer group cylinder 42 TS pocketeye U	0.750	-4	% {}
S100 SILO LEVEL TRANSMITTER U	0.750	5	
Dryer section #2 differential pressure	0.748	5	
BeltSense WINC_R01	0.747	5	

Secondary coarse screen - dilution controller output	0.744	-1	% {}
6th dryer spreader roll current	0.743	-5	% {}
Equipment cooling glycol pump motor current	0.743	5	% {}
TopTurningRollPower_KW	0.743	0	
TopReturnRollPower_KW	0.742	0	
Base ply headbox slice difference measurement U	0.742	-4	
PM Top turning roll current	0.742	0	% {}
Top return roll	0.742	-4	
Slice opening H BP	0.741	-4	% {}
Slice opening H BP	0.740	-3	mm {millimeters}
Slice opening H BP	0.739	-3	mm {millimeters}
Base ply headbox slice difference measurement U	0.739	-4	
S200 SILO LEVEL TRANSMITTER U	0.739	-3	% {}
Equipment cooling glycol pump disch press. Controller output	0.738	5	% {}
DG5 TS water jet tail cutter interlockings U	0.737	-5	mm {millimeters}
Dry end lubrication heating U	0.735	-5	
DG5 DS water jet tail cutter position measurement U	0.735	-4	mm {millimeters}
Actuator maximum temperature	0.734	5	
Hot oil system return flow oil temperature V	0.734	5	
Reel drive drum signals to wincc WINC_R09	0.733	-5	
A/S P.V. & BlowBox SupplyFan AirTemp. Controller output	0.733	-4	% {}
Base ply headbox slice beam heating PV_IN	0.733	-5	
Base ply headbox slice beam heating U	0.733	-5	
Finish ply headbox slice difference measurement U	0.733	-5	
BeltSense WINC_R02	0.732	5	
Fabric trim squirt 1 position U	0.731	-5	in {}
FP machine chest pump Motor current	0.730	-4	% {}
OD actuator back side me	0.730	-5	mm {millimeters}
MO wet end back edge position	0.729	-5	mm {millimeters}
MO actuator back side me (RL)	0.729	-5	mm {millimeters}
MO actuator back side me (SP)	0.728	-5	mm {millimeters}
OD wet end back edge position	0.728	-5	mm {millimeters}
AirTurn SupplyFan after SizePress Air Temp. Controller output	0.727	-4	% {}
LF - refiner feed pump pressure Controller output	0.727	2	% {}
FP blend chest pump Motor current	0.725	-4	% {}
FP machine chest pump - flow to cleaners Controller output	0.725	-4	% {}
BP low vacuum foil box #3 Actual process value	0.724	-5	
6th dryer spreader roll drive power	0.724	-5	
Reel guide roll drive power	0.724	0	
F/P machine chest - level Controller output	0.724	-4	% {}
FP machine chest pump - flow to cleaner's actual setpoint	0.724	-4	
FP machine chest pump - flow to cleaner's actual process value	0.724	-5	gpm {}
Stock flow FP	0.724	-4	gpm {}
Production rate	0.724	-4	

Dryer section #2 steam pressure	0.724	-5	
Dryer section #4 pressure control actual process value	0.720	-5	
Dryer section #4 pressure control actual process value	0.719	-5	
Dryer section #1 pressure control setpoint	0.718	-5	
Dryer section #2 pressure control setpoint	0.716	-5	
Dryer section #4 pressure control actual setpoint	0.716	-5	
Dryer section #3 pressure control setpoint	0.716	-5	PSIG {}
Dryer section #1 steam pressure	0.716	-5	
Dryer section #3 steam pressure	0.716	-5	
GLYGOL EXPANSION TANK LEVEL Measured value output	0.716	5	% {}
PressRun BlowBox SupplyFan AirTemp. Controller output	0.900	-5	% {}
50 Psig header pressure control Controller output	0.894	-5	% {}
PIC-1980_INHG	0.892	1	
MainP.V. BlowBox SupplyFan#3 AirTemp. Controller output	0.887	-5	% {}
Basis weight	0.866	-5	
Basis weight	0.861	-5	g/m2 {}
CurrentBasisWeightTarget	0.861	-5	lbs/rm {}
50 Psig Header Pressure Control Controller Output	0.858	-5	% {}

## Appendix V – Cross-correlation results with time shifts

DESCRIPTION	-5	-4	-3	-2	-1	0	1	2	3	4	5
4th dryer fabric VacRoll current	0.873	0.877	0.879	0.884	0.888	0.900	0.891	0.892	0.892	0.893	0.895
4th dryer top fabric VacRoll drive power	0.870	0.873	0.875	0.879	0.883	0.894	0.888	0.889	0.890	0.890	0.892
SYM-Z hydraulics inflow oil temperature control QPV_IN	0.884	0.885	0.886	0.887	0.888	0.889	0.890	0.890	0.891	0.891	0.892
Tank oil temperature measurement V	0.884	0.884	0.885	0.885	0.886	0.886	0.887	0.887	0.887	0.887	0.887
Bearing lubrication oil temperature return line measurement V	0.866	0.864	0.861	0.859	0.857	0.854	0.851	0.849	0.845	0.843	0.840
4th dryer bottom fabric lead roll A	0.847	0.849	0.851	0.853	0.855	0.861	0.856	0.856	0.856	0.856	0.856
4th dryer bottom fabric lead roll B	0.846	0.848	0.850	0.852	0.854	0.861	0.855	0.855	0.855	0.855	0.856
4th dryer bottom fabric lead roll A drive power	0.843	0.846	0.847	0.849	0.851	0.858	0.854	0.854	0.854	0.854	0.855
4th dryer bottom fabric lead roll B drive power	0.843	0.846	0.847	0.849	0.851	0.858	0.854	0.854	0.854	0.854	0.855
Bearing lubrication tank oil temperature measurement V	0.833	0.835	0.837	0.839	0.841	0.844	0.845	0.846	0.848	0.850	0.852
Bearing lubrication hydraulics inflow oil temperature control QPV_IN	0.842	0.839	0.836	0.833	0.829	0.826	0.822	0.818	0.814	0.811	0.807

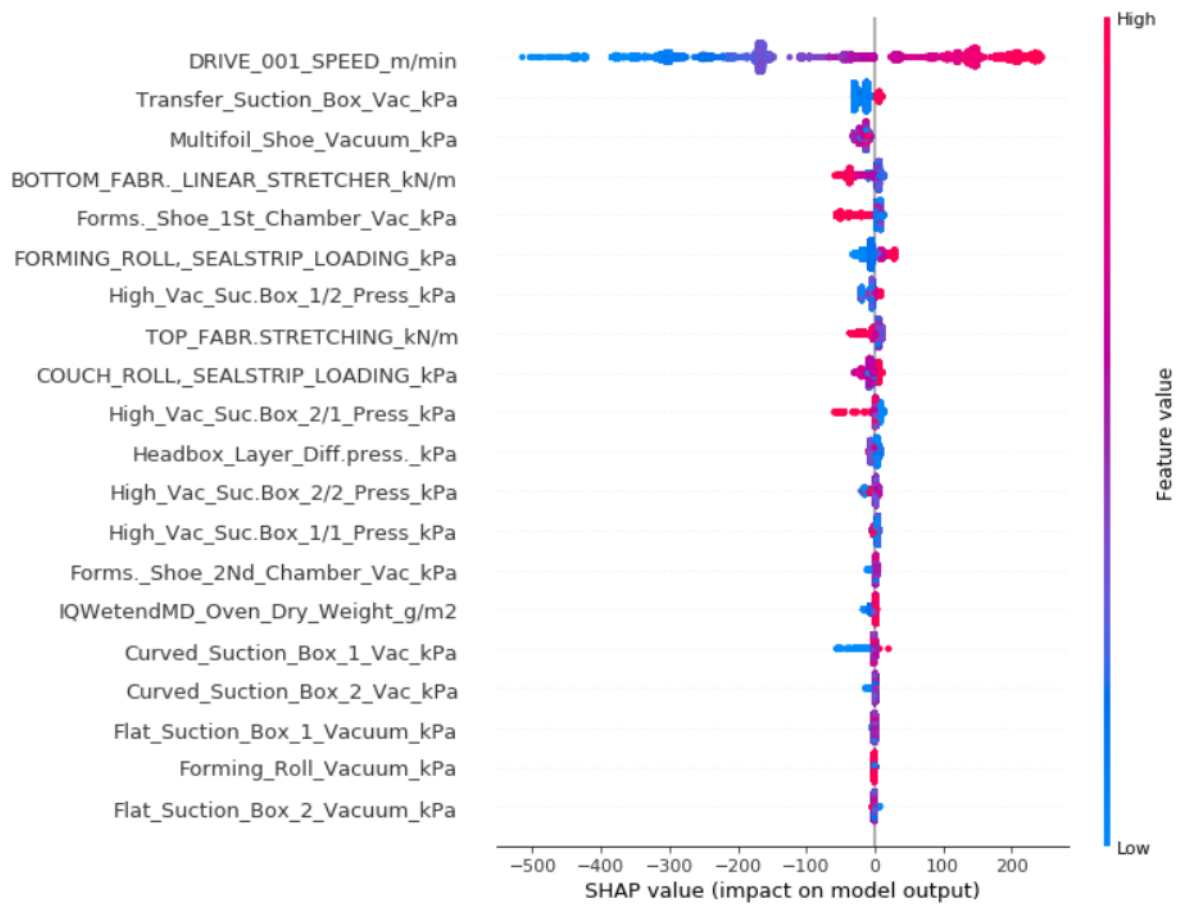
## Appendix VI – Case study: Hyperparameter grid

<b>HYPERPARAMETER</b>	<b>VALUES</b>
<b>LEARNING_RATE</b>	0.05, 0.02
<b>N_ESTIMATORS</b>	500, 1,000
<b>MAX_DEPTH</b>	3, 5
<b>MIN_SAMPLES_SPLIT</b>	10

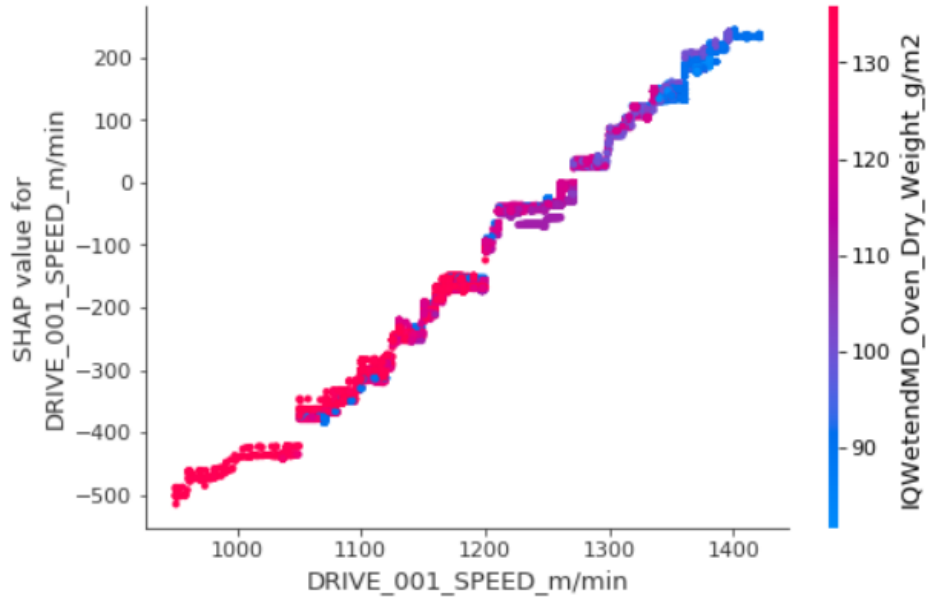
## Appendix VII – Case study: Grid search results

n_estimators	learning_rate	max_depth	min_samples_split	mae_test	mae_train	mae_val	r2_test	r2_train	r2_val
500	0.02	3	10	61.396	29.787	30.256	0.873	0.970	0.969
1000	0.02	3	10	61.95	23.798	24.256	0.873	0.981	0.980
100	0.1	3	2	62.273	29.162	29.698	0.872	0.971	0.970
500	0.05	3	10	62.842	21.713	22.179	0.872	0.984	0.983
1,000	0.05	3	10	63.239	16.618	17.082	0.874	0.991	0.990
1,000	0.02	5	10	65.823	13.054	13.456	0.862	0.994	0.994
1,000	0.05	5	10	66.766	8.962	9.661	0.857	0.997	0.997
500	0.05	5	10	67.141	11.777	12.208	0.854	0.995	0.995
500	0.02	5	10	67.161	17.131	17.464	0.855	0.990	0.990

## Appendix VIII – Case study: Summary plot



## Appendix IX – Case study: Dependence plot



## Appendix X – Case study: Correlating tags of study periods

### STUDY PERIOD 1

Tag	Corr	T_SHIFT	Unit
DRIVE 003 POWER	0.996	0	kW
DRIVE 001 POWER	0.996	0	kW
DRIVE 002 POWER	0.995	0	kW
DRIVE 001 MOMENT	0.995	0	%
DRIVE 002 MOMENT	0.995	0	%
DRIVE 003 MOMENT	0.995	0	%
690V MCC	0.992	0	kW
diff. high speed	0.964	0	%
Machine chest top level	0.963	0	%
Transformer 18 CUR	0.959	0	A
Synth. size dos. btm press	0.958	0	kPa
Gravel expansion production 1	0.957	0	None
Gravel expansion unit 2	0.957	0	None
Valve avg. setpoint	0.957	0	%
Speed ???? setpoint	0.957	0	m/min
Speed ???? setpoint	0.957	0	m/min
Speed ???? setpoint	0.957	0	m/min
Speed ???? setpoint	0.957	0	m/min
Speed ???? setpoint	0.957	0	m/min
Speed ???? setpoint	0.957	0	m/min

### STUDY PERIOD 2

Benton. feed top flow	0.799	0	l/min
Reference value	0.798	0	%
Room temp. in electric room	0.798	0	°C
Reserve concrete pump	0.796	0	Hz
Bentonite standby feed pmp	0.796	0	%
Wastewater level in sewage collection	0.794	0	%
Air temperature in the cable room	0.789	0	°C
Item fil in the. fil. tower	0.787	0	%
Air temperature in the cable room	0.784	0	°C
Air temperature in the cable room	0.780	0	°C
Air temperature in the cable room	0.778	0	°C
Aqua dense mass pump	0.774	0	Hz
Stock to lobemix aquafLOW	0.774	0	%
Reserve concrete pump	0.773	0	%
Collector items for sorting	0.773	0	%
Air temperature in the cable room	0.773	0	°C
Circ.Pipe LF refiner 4 temp.	0.770	0	C
Air temperature in the cable room	0.769	0	°C
Collector items for sorting	0.768	0	%

Thick stock pump Aaqua	0.766	0	%
------------------------	-------	---	---

### STUDY PERIOD 3

DRIVE 001 POWER	0.922	0	kW
DRIVE 002 POWER	0.911	0	kW
DRIVE 003 POWER	0.911	0	kW
690V MCC	0.899	0	kW
PocketVac. V39 web break level	0.882	0	Pa
Transformer 18 CUR	0.877	0	A
Pressure supply mass pias 4 degrees	0.874	0	kPa
PocketVac. V38 Break PeakToPeak	0.871	0	Pa
DRIVE 007 POWER	0.871	0	kW
Pressure distribution on sand 4 degrees	0.862	0	kPa
DRIVE 006 POWER	0.859	0	kW
Current month	0.858	0	l
Ca?k.przep.wód	0.858	0	l/min
Air temperature in the cable room	0.856	0	°C
DRIVE 001 SPEED	0.856	0	m/min
Speed ???? setpoint	0.856	0	m/min
Speed ???? setpoint	0.856	0	m/min
Speed ???? setpoint	0.856	0	m/min
Speed ???? setpoint	0.856	0	m/min
Speed ???? setpoint	0.856	0	m/min

### STUDY PERIOD 4

WATER FLOW, 1.PRESS FELT	0.813	0	l/s
Number of starts	0.793	0	None
YOU WILL RUN IN A CUT TO SD	0.792	0	l/min
Temp. in gearbox turbos 3	0.789	0	°C
Temp. in gearbox turbos 3	0.784	0	°C
Temp. in gearbox turbos 3	0.779	0	°C
Refiner 3 inlet pressure	0.779	0	kPa
Temp. in gearbox turbos 3	0.776	0	°C
DRIVE 013 POWER	0.775	0	kW
DRIVE 014 POWER	0.775	0	kW
DRIVE 015 POWER	0.775	0	kW
Line drives	0.769	0	None
SIZER INTERLOCKINGS DRIVES -> DNA	0.761	0	m/min
Cable room temperature	0.754	0	°C
Headbox turb Diff.Pr Top	0.750	0	kPa
DRIVE 003 POWER	0.750	0	kW
Top ply flow / m	0.750	0	l/s/m
DRIVE 002 POWER	0.750	0	kW
Gram.Obl.Such.Pokr.	0.748	0	g/m2
MultiStock ratio	0.748	0	g/m2

### STUDY PERIOD 5

DRIVE 001 POWER	0.973	0	kW
DRIVE 002 POWER	0.973	0	kW

DRIVE 003 POWER	0.973	0	kW
Headbox recirc. btm cons.	0.972	0	%
Headbox recirc. btm cons.	0.972	0	%
Transformer 18 CUR	0.971	0	A
690V MCC	0.971	0	kW
Stock to lobemix aquaflow	0.968	0	l/min
Stock to lobemix aquaflow	0.968	0	l/min
Stock to lobemix aquaflow	0.968	0	l/min
Blast. 3GR - opening time	0.967	0	s
V38-V39 web on window open time	0.967	0	s
V38-V39 web br window close time	0.967	0	s
V38-V39 web on window close time	0.967	0	s
V38-V39 web br window open time	0.967	0	s
Blast. 3GR - delay	0.966	0	s
Blow 2GR - opening time	0.966	0	s
Blow 2GR - delay	0.966	0	s
IQWetendMD retention	0.966	0	%
IQWetendMD retention	0.966	0	%

### STUDY PERIOD 6

TRIM SQUIRT WATERPRESSURE	0.970	0	kPa
Engine temperature	0.969	0	°C
DRIVE 002 POWER	0.965	0	kW
DRIVE 001 POWER	0.965	0	kW
DRIVE 003 POWER	0.965	0	kW
CI? N 2 GLUE PUMPS	0.964	0	kPa
POWER SUPPLY PUMP 1 SD	0.963	0	Hz
690V MCC	0.963	0	kW
SD FLOW	0.962	0	l/min
PR? DK REF PUMPS ZAS 1 SG	0.962	0	%
PR? DK REF. PUMPS ZAS 2 SG	0.962	0	%
Reference value	0.961	0	%
Reference value	0.961	0	%
POWER SUPPLY PUMP 2 SD	0.961	0	Hz
Form. shoe 1St chamber vac.	0.960	0	kPa
1. DRYER GRPSTATUS FROM DRIVES	0.960	0	m/min
Headbox layer diff.press.	0.959	0	kPa
Headbox layer diff.press.	0.959	0	kPa
White water header temp.	0.959	0	C
Temp.w.cyrkulacyj.30704033	0.959	0	°C

### STUDY PERIOD 7

GPAM set coverage	0.904	0	%
GPAM set from below	0.904	0	%
Pump1sr.p.	0.896	0	Hz
DRIVE 001 MOMENT	0.892	0	%
DRIVE 002 MOMENT	0.891	0	%
DRIVE 003 MOMENT	0.891	0	%

Defoam sizer 1 pump	0.887	0	%
Defoam sizer 1 pump	0.885	0	%
Defoam sizer 1 pump	0.884	0	%
2. PR NIP LOAD LINE PRESSURE FOR	0.877	0	MPa
HEADBOX SLICE OPENING	0.877	0	mm
HEADBOX SLICE OPENING	0.874	0	mm
HEADBOX SLICEOPENING	0.873	0	mm
CI? N 1 GLUE PUMPS	0.863	0	%
Reference value	0.863	0	%
ADHESIVE PUMP 1	0.863	0	Hz
PAM dosing btm pressure	0.862	0	kPa
HEADBOX SLICEOPENING	0.859	0	mm
PAM feed btm flow	0.853	0	kg/min
PAM feed btm flow	0.853	0	l/min

### STUDY PERIOD 8

DRIVE 001 MOMENT	0.985	0	%
DRIVE 003 MOMENT	0.981	0	%
DRIVE 002 MOMENT	0.980	0	%
Form. shoe 2Nd chamber vac.	0.979	0	kPa
High vac. Suc.Box 2/2 press	0.969	0	kPa
PICK UP ROLL SEALSTRIP LOADING	0.966	0	kPa
WATER FLOW, FLATSUCT BOX 1	0.966	0	l/s
Cable room temperature	0.965	0	°C
High vac. Suc.Box 1/1 press	0.964	0	kPa
Vac.Blower 1 inlet 3 press	0.961	0	kPa
Vac.Blower 2 inlet 3 press	0.960	0	kPa
High vac. Suc.Box 1/2 press	0.958	0	kPa
Vacuum into 1 chamber	0.958	0	%
Couch roll 1St chamber vac.	0.958	0	%
Couch roll 1St chamber vac.	0.958	0	%
Couch roll 1St chamber vac.	0.958	0	%
HIRUN DUCT UNDERPRESSURE	0.955	0	Pa
Curved suction box 1 vac.	0.953	0	kPa
Stezen.m. to the towers of SF	0.951	0	%
Headbox recirc. btm cons.	0.951	0	%