Fredrik Heistad, William Andreas Kristensen

Failure Prediction in the Norwegian Power Grid

Master's thesis in MSc in Communication Technology
Supervisor: Poul Einar Heegaard
July 2019

Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

Fredrik Heistad, William Andreas Kristensen

Failure Prediction in the Norwegian Power Grid

Master's thesis in MSc in Communication Technology
Supervisor: Poul Einar Heegaard
July 2019

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

Title: Fault Prediction in Norwegian Power Grid
Students: Fredrik Heistad, William Andreas Kristensen

Problem description:

Advancements in sensor technology and the rollout of smart meters in Norway has drastically increased grid-operators access to information. However, the potential in utilizing this data for Distribution System Operators (DSOs) is still not fully understood. The main objective of this thesis is to investigate to what extent sensor data combined with modern machine learning techniques can be used to improve the support in grid operation, and hence how this will improve the reliability (security of supply) of the Norwegian power grid. Such an algorithm could help grid operators with asset management, reduction of grid outages and improve their quality of service.

The approach taken is to use historical power grid measurement combined with data from other sources (e.g., weather data), run a machine learning method on those sets, and to implement an algorithm for predictive component failures, and if possible suggest root causes. More specifically, the main tasks include:

1. Get an overview of the state of art for machine learning in power grid operation.
2. Collect data from at least one grid operator, and prepare the data set for input to a machine learning method.
3. Select (among methods in task 1.) an appropriate machine learning algorithm and implement a prediction algorithm.
4. Run the algorithm (or several algorithms) from task 3. on the dataset in task 2.
5. Discuss the interpretation of the outcome of the prediction algorithm and it's usefulness in grid operation support.

Responsible professor: Poul E. Heegaard, IIK
Supervisor: Romina Muka, IIK

Abstract

The development of the smart grid has led to an increase in the number of sensors and smart meters installed in the power grid. These devices make large quantities of data available to grid operators. At the same time, advancements in the field of machine learning have enabled powerful tools to provide valuable insight from big data. In predictive maintenance, machine learning is used as a tool to predict components failures before they happen.

This thesis introduces a process to determine whether data from sensors in the power grid can be used to predict grid failures. The thesis is a contribution to research on the potential of utilizing grid information for operation support in the power grid. We present our findings on how machine learning and predictive maintenance may be applied in the power grid domain. Further, we propose methods for building machine learning models for prediction of component failures in substations.

Our research includes a literature study and an experiment. As part of the experiment, we have gathered grid data from a major Norwegian grid operator. Our experiments consist of two separate parts with different approaches to build prediction models for failures in substations. In the first approach, we present a supervised learning technique to predict exact future outcomes. In the second approach, we use unsupervised learning techniques for building models able to detect anomalous sensor measurements. Our best performing model detected a statistically significant number of anomalies, prior to the time of failure, in two of the eight failures investigated.

Sammendrag

Utviklingen av smartgrid har ført til en enorm vekst i antallet sensorer og smarte målere i strømnettet. Disse enhetene tilgjengeliggjør store mengder data. Samtidig har det de siste årene skjedd store fremskritt innen fagfeltet maskinlæring. Disse fremskrittene har gitt oss kraftige verktøy for å hente verdifull informasjon ut av store datamengder. Innenfor prediktivt vedlikehold brukes nettopp maskinlæring til å predikere komponentfeil før de inntreffer.

I denne oppgaven introduserer vi ulike metoder for å utnytte sensordata, sammen med værdata, til å predikere feil i det norske strømnettet. Oppgaven er et bidrag til forskning på utnyttelse av sensordata for å bedre driften av strømnettet. Mer spesifikt ser vi på ulike metoder for å bygge maskinlæringsmodeller for prediksjon av komponentfeil i nettstasjoner.

Gjennom oppgaven har vi studert eksisterende forskning på emnet og gjennomført et eksperiment. I eksperimentet samlet vi data fra et av Norges største nettselskaper, og brukte denne til å utvikle tre maskinlæringsalgoritmer. Implementasjonen av den tekniske løsningen i eksperimentet er delt opp i to sidestilte fremgangsmåter. I den første fremgangsmåten benyttet vi veiledet læring til å bygge en modell som kan predikere feil i nettstasjoner. I den andre fremgangsmåten bygger vi en modell ved hjelp av ikke-veiledet læring, for å oppdage målinger som avviker fra normaltilstanden.

Oppgaven forklarer den tekniske implementasjonen av algoritmene, etterfulgt av en diskusjon knyttet til antakelser, begrensninger og hensyn vi har måttet ta under utviklingen. Vi diskuterer også hvordan dette har påvirket resultatene og substansen i oppgaven. Vår beste modell oppdaget et statistisk signifikant antall avvik fra normaltilstanden for to av de totalt åtte feilene som ble studert.

Preface

This Master's Thesis is written as part of the Communication Technology Program at the Department of Information Security and Communication Technology, Norwegian University of Science and Technology.

First and foremost, we would like to express our sincere thanks and appreciation to our responsible professor Poul Einar Heegaard for his weekly guidance and helpful advice throughout the work with this thesis. Also, we would like to express our gratitude for the guidance from our supervisor Romina Muka. They helped us define our subject for this thesis. Additionally, we wish to thank our DSO for their collaboration and willingness to both share data and devote time and effort to help us.

Also, we wish to thank each other. We have learned from one another, and from working as a team during this thesis. We have had long discussions, and it has not always been easy. However, our friendship helped motivate us, and has grown even stronger in the process. Finally, thanks to all the fantastic people we have met during our five years in Trondheim. We hope you enjoy your reading.

Fredrik Heistad, William Andreas Kristensen
Trondheim, July 1st 2019

Contents

List of Figures **ix**

List of Tables **xi**

List of Acronyms **xiii**

1 Introduction **1**

1.1 Motivation . . . 1
1.2 Research questions . . . 3
1.3 Methodology . . . 3

2 Background **7**

2.1 The Norwegian Power Grid . . . 7
2.1.1 Infrastructure . . . 7
2.1.2 Security of Supply and Instantaneous Balance . . . 10
2.1.3 Peak Load . . . 10
2.1.4 The Norwegian Power Industry . . . 11
2.1.5 Failures in the Grid . . . 12
2.2 Development of the Smart Grid . . . 14
2.2.1 Renewable Energy . . . 14
2.2.2 Grid Utilization and Flexibility . . . 15
2.2.3 What is the Concept of Smart Grid? . . . 16
2.2.4 Smart Meters and Information Availability . . . 17
2.3 Predictive Maintenance . . . 18
2.3.1 Maintenance Management . . . 18
2.4 Machine Learning . . . 21
2.4.1 History and General Idea . . . 21
2.4.2 Artificial Intelligence vs Machine Learning vs Deep Learning 23
2.4.3 Supervised vs Unsupervised Learning . . . 24
2.4.4 Data Preprocessing: Providing useful data to the algorithms 27
2.4.5 Performance Evaluation . . . 30
2.4.6 Libraries . . . 31

3 Related Work **33**

4 Technical Approach **39**
4.1 Data Collection . . . 39
4.2 Data Characteristics . . . 41
4.3 Selection of Machine Learning method . . . 43
4.4 The Supervised Implementation . . . 44
4.4.1 Data Preprocessing . . . 45
4.4.2 Predictions . . . 46
4.5 The Unsupervised Implementation . . . 47
4.5.1 Data Preprocessing for Unsupervised Learning . . . 47
4.5.2 Parameters . . . 50
4.5.3 Training the Models . . . 52
4.6 Performance Metrics of Prediction Models . . . 53

5 Results and Discussion **57**
5.1 Results . . . 57
5.1.1 Supervised results . . . 57
5.1.2 Results from Unsupervised Implementations . . . 58
5.2 Interpretation of Results . . . 65
5.2.1 Assumptions . . . 69
5.2.2 Substance . . . 70
5.3 Review of Research Questions . . . 71
5.4 Challenges and Limitations . . . 73
5.4.1 Process of Data Collection . . . 73
5.4.2 Quality and Quantity of Collected Data . . . 74
5.5 Evaluation of Methodology . . . 75
5.5.1 Conversation with the Distribution System Operators . . . 75
5.5.2 One Complex Model vs Many Separate Models . . . 76
5.5.3 Considerations of Technical Implementation . . . 77

6 Conclusions and Further Work **79**
6.1 Conclusion . . . 79
6.2 Future Work . . . 80

References **81**

Appendices

A Python Code **87**

List of Figures

1.1 Visualisation of the steps included in our research methodology. 3

2.1 The power grid enables electricity to flow from production to consumers. 8
2.2 A simplified topology of the Norwegian power grid and its operators. The voltages typically carried in the different levels is also included. 9
2.3 Relationship between frequency and the load–generation ratio.[DO18] . 10
2.4 The most common root causes of failures in the high-voltage distribution grid (1-22 kV). The numbers are gathered from the Fault And Supply Interruption information Tool (FASIT) report [voe18]. 13
2.5 World electricity generation by power station type. Source: DNV GL Energy Transition Outlook 2018 15
2.6 Illustration of the smart grid concept [Bar]. 16
2.7 The Bathtub Curve [Suh15] 19
2.8 Relationship between the fields of Artificial Intelligence (AI), Machine Learning (ML) and Deep Learning (DL). 23
2.9 Random forest illustrating the majority voting of decision trees. 24
2.10 Simplified illustration of supervised and unsupervised learning. 25
2.11 Simplified illustration of a one-class SVM. 26
2.12 Underfitted, good fit, and overfitted machine learning model [Anu18] . . 31

3.1 Wildlife fault that stressed connector [RBC+09] 34
3.2 Anomalies following wildlife stress, prior to outage [RBC+09] 34

4.1 Visualisation of the main tasks included in the technical approach. . . . 39
4.2 Process diagram. 43
4.3 The three algorithm approaches conducted, one supervised and two unsupervised. 44
4.4 Sliding Window 46
4.5 Example of training and testing data used in second unsupervised approach. 48

4.6 The timeline shows the intervals used for training sets and test set with respect to the time of failure, in the second unsupervised approach. The time of failure is not included in the test set. The example corresponds to data from *S1* on 27.10.2018. 49
4.7 The difference between a large and small gamma parameter 51
4.8 Confusion matrix used for visualising model performance 53

5.1 The confusion matrix for the supervised two-class classification. 58
5.2 Distribution comparison plot with data for all four test sets. 59
5.3 Predictions for *S1* and *S2*, 27.10.2018. 60
5.4 Predictions for *S1* and *S2*, 01.01.2019. 60
5.5 Distribution comparison plot with data for all four training and test sets. 63

List of Tables

2.1 Categorical text data 28
2.2 One-hot encoded variables 28

3.1 Table summarizing various related articles 38

4.1 Overview of the data collected from substations 41
4.2 Training set and test set for the first unsupervised approach. 47
4.3 Failure times, training sets, and testing sets used in the second unsupervised approach. 49
4.4 Parameters used for the One-Class SVM 50

5.1 The table presents the results for the first unsupervised approach from the classifications of observations prior to each failure in the test sets. . 61
5.2 The table presents the results of the hypothesis testing for the first unsupervised approach. 61
5.3 The table presents the results from the second unsupervised approach. . 64
5.4 Hypothesis Test Results for the second unsupervised prediction model, using minute values. 64
5.5 Obtained *p-value* in hypothesis test results for the second unsupervised prediction model, using minute values. With H_0 being that 5% of the observations prior to a failure is classified as anomalies, and the *p-value* is the probability of finding the observed, or more extreme, results when the null hypothesis (H_0) is true. 65

List of Acronyms

AI Artificial Intelligence.

ANN Artificial Neural Network.

CENS Cost of Energy Not Supplied.

CINELDI Centre for Intelligent Electricity Distribution.

CV Cross Validation.

DL Deep Learning.

DSO Distribution System Operator.

FASIT Fault And Supply Interruption information Tool.

GW Gigawatt.

HV High Voltage.

Hz Hertz.

KNN K-Nearest Neighbors.

LV Low Voltage.

ML Machine Learning.

MTBF Mean Time Between Failure.

MTTF Mean Time To Failure.

NVE the Norwegian Water Resources and Energy Directorate.

OCC One Class Classification.

One-Class SVM One-Class Support Vector Machine.

PdM Predictive Maintenance.

PMU Phasor Measurement Unit.

PQA Power Quality Analyser.

RF Random Forest.

SVM Support Vector Machine.

TSO Transmission System Operator.

VRE Variable Renewable Energy.

Chapter 1 Introduction

The purpose of this thesis is to determine whether data from sensors in the power grid can be used to predict grid failures. Why should prediction of grid failures be of interest to stakeholders in the power grid? Below we present the motivation, research questions, and methodology for this thesis.

1.1 Motivation

Electricity is the backbone of the modern world, and high security of supply is a prerequisite for well-functioning societies. However, urbanization, population growth and increased demand for electricity are leading to an increased load on the power grid. Norway's power grid has gradually been built and maintained over the last century. This means that components in the grid, such as circuit breakers, transmission lines, transformers and metering equipment vary in both age and quality. Some of these components are crucial for the operation of the grid and needs to be replaced and maintained periodically.

Advancements within information and communication technology, are used to upgrade the current grid to the so-called smart grid. Smart grids are commonly referred to as next-generation power systems. They apply sensing and measurements, two-way communication and power system automation. Smart grids will lower costs, save energy, improve security of supply, operability, and reliability, with automated control and modern technologies [EYSKBL17][VCG10].

Statnett, the Norwegian Transmission System Operator (TSO), states that there will be comprehensive changes to the power grid in the coming years. One of the four main areas of development in their action plan towards 2021 is an improvement of decision support systems and increased automation in the system operations [Sta17]. Furthermore, Statnett claims that unavailability of grid components may lead to substantial costs for society in the form of increased market costs. However, well-coordinated outages limit the consequences. Also, the grid operators are aware of the potential in embracing technology. As a result, many of the operators are now installing a significant number of sensors in the grid, to gather vast amounts of data. However, there is still uncertainty of exactly how to leverage this data effectively.

Many industries are researching how massive amounts of data can be used to generate actionable insight and clinical decision support. Investigating whether one can capture recurring patterns to predict component breakdowns can be of interest for grid operators. This will change their maintenance management from being primarily reactive to become more proactive. We believe that data from the sensors can be used to perform predictive maintenance and that predictive analytics may be used to understand the likelihood of a component failure within a certain amount of time. This way, grid operators can send field workers to perform live checks of the component, and potentially repair or change it before failure occurs.

As previously mentioned, Distribution System Operators (DSOs) gather massive amounts of data from sensors deployed in the power grid. At the same time, machine learning technologies have matured and are used to provide better operating support in many industries. Today machine learning is used in applications such as fraud detection, personal assistance and self-driving cars. Hence, with the increase in available data sources for the DSOs, we intend to explore the potential of using historical and real-time sensor data to provide insights which can support grid operation.

1.2 Research questions

We wish to investigate to what extent grid companies can utilize their sensor data to identify the likelihood of a component failure in the grid. Identification of vulnerable components could help grid companies to intervene before customers lose their power due to component breakdown. Such a solution might be economically beneficial for grid operators, and ensure more reliable access to electricity for customers.

Our research questions are:

1. Why should machine learning and predictive maintenance be of interest to Norwegian grid companies?
2. Which external data sources can be combined with grid data to increase insight on the distribution grid?
3. What is the potential of using sensor data and machine learning techniques for predictive maintenance in operation of the Norwegian distribution grid?

1.3 Methodology

This section describes the research methodology used for providing answers to our research questions. Figure 1.1 visualizes the methodology broken up into tasks which will be presented throughout the thesis. The tasks of "Background research", "Literature study" and "Semi-structured interviews with DSOs" are part of an iterative process for defining a hypothesis, while the remaining tasks are executed sequentially. The structure of the thesis follows the methodology.

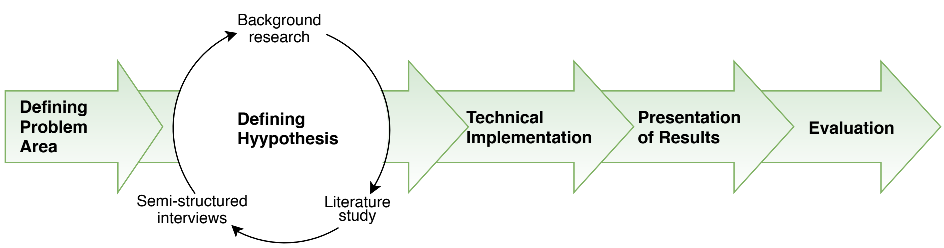

Figure 1.1: Visualisation of the steps included in our research methodology.

In the Background, we study some of the topics relevant for our objective. First, we introduce the Norwegian power grid in terms of physical infrastructure before discussing the industry as a business. Further, the concept of smart grid is discussed to gain an understanding of the motivation and goals for developing a smarter power grid. We then move on to the more technical part of the background, when discussing predictive maintenance and machine learning.

To gain insight for answering our research questions, we present an overview of the state of the art for machine learning in power grid operation. This insight is gained through a comprehensive literature review. Various papers from both Norwegian and international researchers are studied. From the research-papers discussed in Related Work, we have learned approaches, struggles and thoughts on suggestions for future work. The literature review was carried out simultaneously with ongoing conversations with two DSOs. In an iterative process of literature review and semi-structured interviews with these DSOs, we refined our research questions to a hypothesis. The hypothesis states that it is possible to use data from the power grid to perform predictive maintenance on substations. Through conducting experiments with data obtained from DSOs, we further advance to test this hypothesis.

In the next part of our methodology, the Technical Approach, we set up an experiment including a technical implementation where the goal is to build a machine learning model able to predict faults occurring in substations. This chapter describes our process of collecting data from the DSOs and exploring characteristics of this data. Taking into consideration what we learned from the literature review, as well as the characteristics of the received data, we select appropriate implementation techniques for building a prediction model. Further in the Technical Approach, the chosen machine learning methods are presented and described in detail. Finally, metrics for measuring the performance of the prediction models are discussed, taking into account case-specific matters. To the best of our knowledge, there have been no previous projects regarding fault prediction on the Norwegian power grid, including an attempt of technical implementation. However, a paper from 2018[ATHU18] states that with the amount and precision of data recorded in the power grid today, prediction should be feasible.

After describing the technical implementation, the achieved results are presented in Results and Discussion, including performance metrics of the prediction models. The results will be investigated and evaluated in terms of validity and limitations in order to answer whether or not the defined hypothesis holds. After reviewing the hypothesis, the chapter will move on to discuss more general limitations and challenges of this research. A discussion of the usefulness of prediction models such as the one presented, as well as how our work can be used for further research is next. Our experiences will also be used to discuss how Norwegian grid companies can adapt in order to position themselves better for utilizing data from the power grid. The chapter will conclude by revisiting our research questions, incorporating what we have learned to provide answers to these.

To summarize the structure of the thesis:

- Chapter 2 discusses the Norwegian power grid and maintenance tasks, what machine learning is and how it can be used to support operation of the grid.
- Chapter 3 presents the most relevant related work found during our literature review
- Chapter 4 describes our technical implementation, as well as the reasoning for choosing the methods which were used.
- Chapter 5 presents and interprets the results of the experiments, before discussing the substance of these in relation to our research questions.
- Chapter 6 provides the concluding remarks and a discussion about future research on the topic we have researched.

Chapter 2
Background

This chapter provides the theoretical background of the Norwegian power grid and explains the concepts of smart grid, machine learning and predictive maintenance. We explain why our field of research is relevant, for the reader to gain an understanding of why we have chosen our respective research questions.

2.1 The Norwegian Power Grid

This Section will be focused around the infrastructure of the Norwegian power grid, its operation and stakeholders, as well as typical failures occurring in today's grid.

2.1.1 Infrastructure

Almost all parts of a modern society depend on a well-functioning power system. The power grid enables electricity to flow from producers to customers, and is a key infrastructure and the backbone of the power system [Nor19]. Figure 2.1 shows how producers connect to one common grid from different topological locations, and that Norway is a net exporter of electricity. The Norwegian electricity grid may be divided into three main levels between production and consumption. The three layers are the transmission grid, the regional grid and the distribution grid. Figure 2.2 visualizes the topology of the infrastructure and how the different parts are defined in terms of voltages.

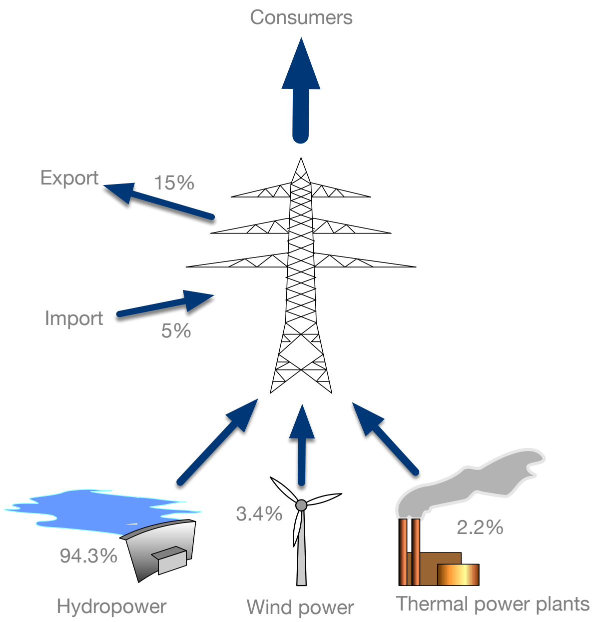

Figure 2.1: The power grid enables electricity to flow from production to consumers.

Power Generation

Most of Norway's electricity production originates from renewable energy sources, such as hydro and wind power. In fact, 96% of the electricity generation in Norway comes from the 1660 hydroelectric power plants in the country [Nor19]. The location of production sites is heavily based on the accessibility of resources. As resources are unevenly spread out across the country, the electricity grid is essential for delivering power to consumers often located far away from production sites.

Transmission Grid

The transmission grid, also known as the central grid, carries high voltage, usually between 300 to 420 kV. The transmission grid can be viewed as the motorway of the power system. Througout the country, the transmission grid branches out to lower voltage regional and distribution grids. Statnett, a state-owned enterprise, is the designated TSO in Norway. As the TSO, they are responsible for maintenance and extension of the transmission grid, which has a total length of about 12 500 km.

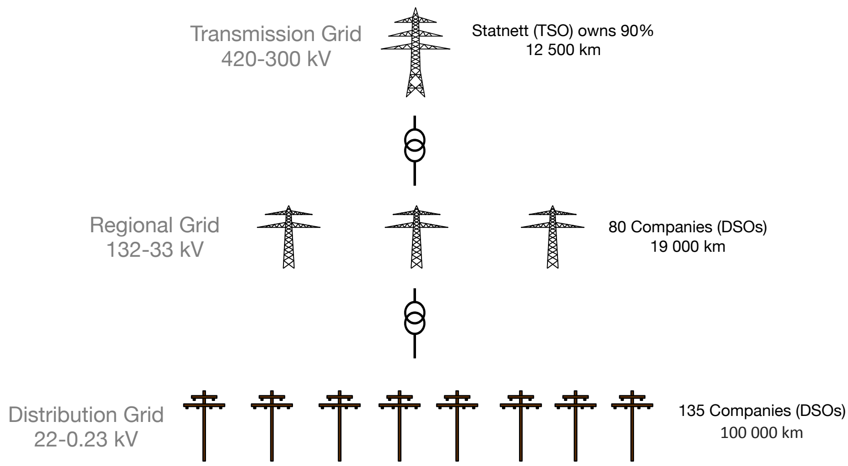

Figure 2.2: A simplified topology of the Norwegian power grid and its operators. The voltages typically carried in the different levels is also included.

Regional Grid

The regional grid is the link between the transmission and distribution grid, and is operated by DSOs, companies given concession by the government [Lov]. Voltages carried in the regional grid are in the range from 33 to 132 kV. Endpoints connected directly to the regional grid may include smaller production facilities and power-intensive manufacturing or customers from the petroleum industry. The regional grid has a total length of about 19 000 km, and the operation is split between 80 companies [nve].

Distribution Grid

The distribution grid is operated by local DSOs. The distribution grid supplies smaller end-users such as households. This part of the grid carries voltages between 230 V and 22 kV. The distribution grid can be further divided into low and high voltage segments, split at 1 kV. The length of the distribution grid is 100 000 km, and it is operated by 135 companies. The final voltage transformation before a power line reaches a household, the transformation down to 230 V, happens in substations. In this thesis we study data provided from sensors installed in such substations.

2.1.2 Security of Supply and Instantaneous Balance

The International Electrotechnical Commission (IEC) has defined Security of electricity supply as the "ability of an electric power system to provide electric power and energy to end-users with evaluation of existing standards and contractual agreements at the point of supply" [IEC19]. This means that security of supply is about maintaining a stable supply of electricity at an acceptable cost [sin].

The fact that electricity cannot be stored means that there must at all times be a balance between power generation and consumption. This balance is known as the instantaneous balance, and needs to be maintained at an equilibrium. A badly maintained instantaneous balance may lead to damage in components, which again may lead to power outages. Whenever cars are charging or factories are producing goods, the consumed electricity has to be generated simultaneously with the consumption. Statnett, the TSO in Norway, is responsible for maintaining the instantaneous balance at all times. The system frequency is a measure of the balance, and the nominal frequency is 50 Hertz (Hz). Frequency quality can be measured using deviations, expressed as the number of minutes outside normal variation range of 49.9-50.1 Hz [Ene19].

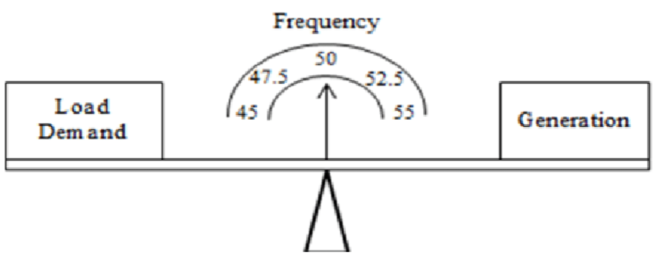

Figure 2.3: Relationship between frequency and the load–generation ratio.[DO18]

2.1.3 Peak Load

Load on the power system is an indication of how much electricity is consumed at a specific point in time. This load fluctuates and has a clear correlation with the temperature. However, the systems peak load has had a distinct trend in the last 30 years. The maximum load on the system in 1990 was 18.42 Gigawatt (GW). In 2016 the maximum load reached 24.49 GW. There has been a 33% rise in the peak load on the system since 1990. There has also been a general rise in consumption, a trend expected to continue in the future [Ene19]. Heavier load on the system may result in faster wear of its components, thus more frequent component failures.

2.1.4 The Norwegian Power Industry

In order to understand how DSOs operate, it is necessary to have some knowledge about stakeholders associated with the power grid, and their respective interests. The regional and distribution grids are operated by local DSOs, who are given concession by the Norwegian Water Resources and Energy Directorate (NVE) [Lov]. The operators act like monopolists, as having multiple power grids in the same area would not make sense concerning the cost of infrastructure. From a socio-economic perspective, it is desirable that the DSOs operate in a way that leads to reliable delivery of electricity at low prices, while maintaining sustainable development of the environment [BBM15].

To ensure that the power grid is operated as efficiently as possible from a socio-economic perspective, DSOs have to comply with regulations dictated by NVE. Regulations from NVE include price ceilings and Cost of Energy Not Supplied (CENS)[TL13]. CENS is compensation for non-delivered energy during an outage, and works as an incentive for DSOs to reduce the number of outages and the duration of these. Grid companies are fined based on the predicted amount of electricity not delivered [TL13] [Tje16]. Additionally, politics and reputation are pushing the DSOs towards environmental awareness [BBM15].

Norwegian DSOs are partially owned by the local municipalities from the area which they serve. The DSOs primary objective is to maximize profit by ensuring operational efficiency so that dividends can be paid to their owners, while also meeting the regulatory requirements set by NVE. Regulations on price ceilings are set based on the historical performance of DSOs, and the companies are compared to the average DSO in Norway [PMG04]. This means that even though operators do not have any direct competitors in their area, one could still argue that they are competing, in terms of outperforming each other on operational efficiency. Therefore, the process of increasing profits for a DSO is primarily about increasing the effectiveness of their operation more than other DSOs.

Traditionally, main stakeholders in the grid have been grid companies, regulators, and equipment suppliers. However, the emergence of new instrumentation and more access to data allows for third-party stakeholders to provide support for the operators. Established software companies, as well as new and innovative electricity providers, such as Tibber, using intelligent software to manage electricity, are likely to become significant stakeholders in the power grid [Inn19].

2.1.5 Failures in the Grid

The electricity grid is considered a critical infrastructure. Interruptions in the power supply may have serious consequences for end-users. The grid must be able to cope with the variability of demand and varying voltage quality and earth faults. We introduce earth faults, and explain why the industry has expressed a request for software able to identify and categorize earth faults based on measurement data. Additionally, we introduce the statistics of faults and disturbances in the Norwegian grid.

Fault Statistics

In the Norwegian power grid, there is a mandatory responsibility for DSOs and the TSO to report faults and disturbances through a national reporting system called FASIT [voe18]. The report must specify the type of incident, time and duration, number of affected phases, voltage level, as well as plausible root cause and contributing causes. This reporting gives a statistical basis for national reports on the frequency of faults and disturbances in the Norwegian grid.

From the 2017 FASIT report [voe18], we can learn that the average Norwegian end-user experienced 1.6 short and 1.7 longer power outages during that year. Longer power outages are defined as outages lasting for more than three minutes. The number of events and consequences is far larger in the high voltage distribution grid (1-22 kV) with 8 672 events, compared to the transmission and regional grid (33-420 kV) where only 459 events were registered. The FASIT report states that in the distribution grid the major cause of errors is "surroundings" with 52.8%. Technical equipment is the major root cause in 12.3% of the failures, and in 22.9% of the cases the root cause is non-defined. Surrounding causes include birds, thunderstorms, vegetation and wind. Figure 2.4 provides an overview of the major cause of errors in the distribution grid.

Figure 2.4 shows the how the surrounding causes are distributed when split into more specific causes, such as birds, vegetation, thunderstorms and wind. The root cause responsible for most failures in the high-voltage distribution grid is technical equipment. One may discuss if surrounding causes, such as rain and wind, really should be considered as surrounding causes. If a component fails due to heavy rain, it is difficult to know if the operators categorize the outage as a component failure due to rain. The same goes for operational stresses. Where these lines are drawn are to our understanding up to the various grid operators. Hence, this data should be taken with a grain of salt.

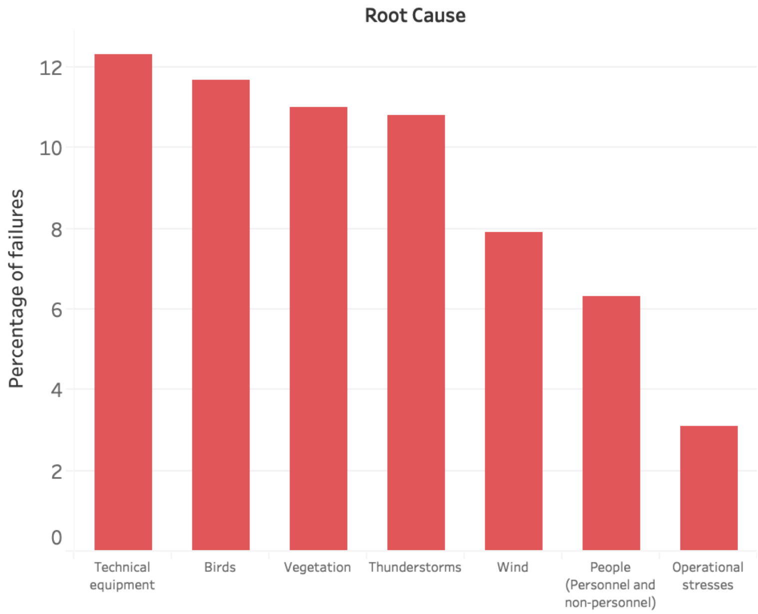

Figure 2.4: The most common root causes of failures in the high-voltage distribution grid (1-22 kV). The numbers are gathered from the FASIT report [voe18].

The regional- and transmission grid generally contains equipment providing the TSO with a relatively high degree of insight and control of the behaviour of their networks, through sensors and remotely controlled switches [KS15]. On the other hand, the distribution grid, where nearly 95% of all faults happen, is equipped with a limited amount of sensors to give the distributors insight into the current status of the grid. According to [KS15], a paper from 2015, Norwegian DSOs are “blind and happy – until the customer calls". Thus, detecting, localizing and repairing faults in the low-voltage network ($<$ 1kV) often takes more time than necessary.

In 2017 the average recovery time on breaches affecting the end-user was 1 hour 22 minutes, according to a NVEs document on interruption-statistics [voe18]. Restoration of physical components is a time-consuming task with manual labour required. Thus, DSOs should be interested in automating tasks of monitoring and restoring components in the grid. Utilization of historical and real-time data for increasing insight have helped many industries in a move towards a more proactive form of operation. The combination of new technology and the increasing cost of failures[MHH18] should be reasons for DSOs to research ways in which they can operate in a more proactive manner.

2.2 Development of the Smart Grid

As this thesis should be regarded as a contribution towards the development and implementation of a smarter power grid, a solid understanding of the underlying motivation and goals for the smart grid is necessary. This Section discusses topics which provide the foundation of smart grid, before diving into more concrete details on the implementation of sensors and controllers, and the potential these bring through the availability of new information.

Concerning the challenge of global warming, greenhouse gas emissions from the burning of fossil fuels like coal and oil are the biggest issue [NAS]. Thus, reducing the consumption of energy from fossil sources is a crucial part of responding to the threat of climate changes. Before this reduction can happen without the global economy suffering, alternative energy sources needs to be available. The worlds increasing awareness on climate changes and environmental issues leads to a greener energy consumption, relying heavier on electricity than before. Development of what has been termed the smart grid has emerged as a consequence of this trend.

2.2.1 Renewable Energy

Investments in research and development of renewable energy have been steadily growing over the last 15 years [ES18]. Technology for utilizing renewable energy sources like wind and solar power are continuously has improved and become more cost and energy efficient [Laz].

Renewable energy sources like wind and solar energy differ from fossil energy sources in many ways. First of all, while coal and oil can be stored, transported, and burned at any chosen time. However, wind and solar energy have to be converted into electricity to be utilized. Hydropower, the primary source of electricity in Norway, has a similar advantage as fossil fuels of being flexible in the sense that water can be stored in reservoirs until electricity is needed. Because of this difference, wind and solar energy are referred to as Variable Renewable Energy (VRE).

In order to comply with international agreements on emission reductions, countries need to plan and facilitate for renewable energy sources. This is leading to the electrification of many sectors which have traditionally been powered by fossil fuels. Between the years 2004 and 2017, electricity consumption in Norway has varied from 121.9 to 134.3 TWh [SSB18]. In May 2019, NVE published a report with forecasts for electricity consumption in Norway, suggesting a 22% increase from 130 to 159 TWh between 2015 and 2040. Contributions to this increase will mainly come from the electrification of transport and the petroleum industry, as well as establishments of data centers.

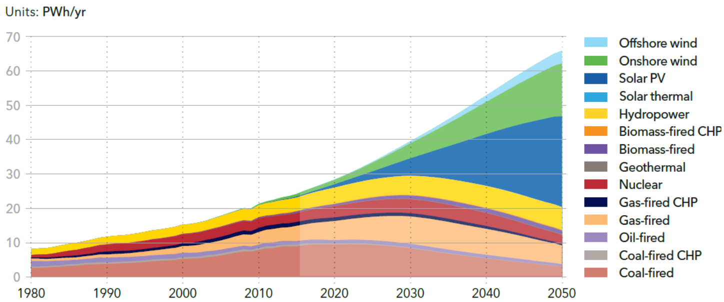

Figure 2.5: World electricity generation by power station type. Source: DNV GL Energy Transition Outlook 2018

2.2.2 Grid Utilization and Flexibility

NVE publishes a yearly report about the status and forecasts of the power grid infrastructure. In the report from 2018, it is stated that more than 50% of the grid infrastructure investments for the next decade will be related to either the increasing consumption or unsatisfactory technical condition. While some parts of the grid still have available capacity capable of handling the increasing maximum load, other parts are already operating close to its load capacity. [LO].

Because of the natural variation in electricity consumption throughout the day, with demand peaks in the morning and afternoon, flexibility in production volume is needed. In Norway, this has historically been solved mainly through the flexibility that is brought by hydropower. With the introduction of VRE sources for production, both in large and small scale, the requirements for the power grid will have to change in order to maintain the ability of efficiently utilizing production capacity [AK]. Solar panels are not going to produce energy during the night, and wind turbines need wind to produce electricity. Thus, to fully take advantage of these renewable energy sources, the grid needs to become more flexible. One way of defining this flexibility is that "The concept of flexibility describes the capability of the power system to maintain balance between generation and load under uncertainty" [HA17].

As the cost of upgrading the grid infrastructure is extremely high, looking at alternative solutions for fulfilling future grid capacity requirements becomes important from a socio-economic perspective. Peak-shaving of load through automated demand response tools is one of the concepts that can drastically reduce or delay the need for infrastructure investments [PD11]. By incentivizing customers to shift their consumption through time-varying power prices, one hopes to reduce the maximum load on the grid. Other fields of research, like energy storage, is also expected to provide effective tools contributing to the mentioned grid flexibility[AK].

2.2.3 What is the Concept of Smart Grid?

Taking into consideration the effects of the mentioned circumstances and how the power grid needs to change in the coming years, the concept of smart grid has emerged as a solution to the grid needs to be developed.

Various definitions of the smart grid concept exist. However, the one used in [GSK+11] provides a useful summary of how the smart grid solves different challenges. Here the concept is explained as "a modern electric power grid infrastructure for enhanced efficiency and reliability through automated control, high-power converters, modern communications infrastructure, sensing and metering technologies, and modern energy management techniques based on the optimization of demand, energy and network availability, and so on."

When communicating sensors and controllers are installed on top of existing grid infrastructure in large scale, the potential of building software for supporting the operation of the grid changes. Smart grids, commonly referred to as the next generation electric power system, apply real-time monitoring, networking, and control technologies. The smart grid is said to lower cost, save energy, improve security, operability, and reliability, with an integration of renewable and alternative energy sources, through automated control and modern technologies [EYSKBL17, VCG10].

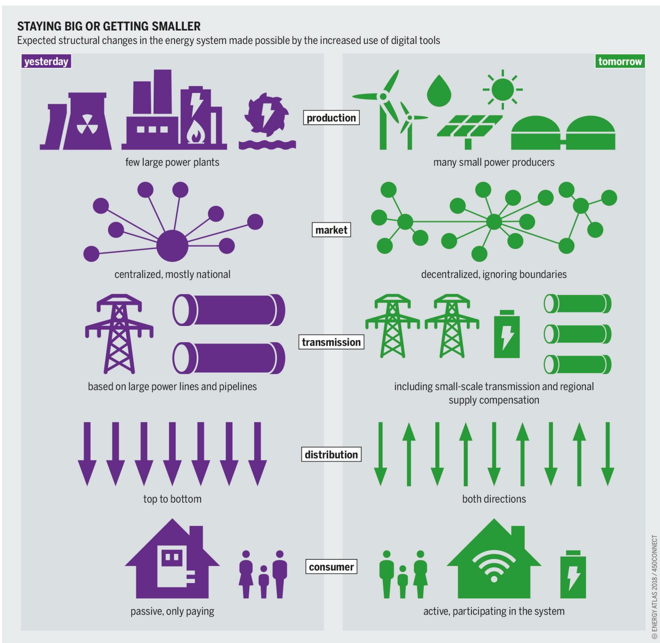

Figure 2.6: Illustration of the smart grid concept [Bar].

2.2.4 Smart Meters and Information Availability

In the Norwegian low voltage distribution grid, there are few sensors installed. However, the rollout of smart meters to every Norwegian household enables sophisticated measurements of volumes and patterns in electricity consumption. Implementing solutions for increasing insight into the state of the power grid is a prioritized task for many DSOs. With the advancements of smart grid and the installation of smart meters in every Norwegian household, the availability of information from the power grid will grow significantly in the coming years. Although information availability itself is not going to improve the power grid significantly, it creates opportunities for building software that takes advantage of this information to provide tools for supporting the operation.

Gungor, Lu and Hancke [VCG10] discusses the impact of increased information availability from the power grid. They expect low-cost monitoring and control enabled by sensor technology to become essential to maintain safety, reliability, and efficiency in the power grid. Sensor nodes installed on critical components will enable real-time monitoring of the grid on a different level than what has been possible before. By analyzing sensor data of more substantial quantities than what has been done before, new insight is likely to be found. Gungor et al. [GSK+11] further expect that the negative impact of equipment failures, capacity limitations, and natural accidents, causing disturbances and outages in the power grid, can largely be avoided by monitoring along with software for diagnostics and protection.

2.3 Predictive Maintenance

There is great interest, as well as large investments, towards the field of Predictive Maintenance (PdM). The purpose of this section is to provide a deeper understanding of the concept of predictive maintenance. Some failures in the power grid may be considered impossible to predict, due to external instantaneous events, such as bird-related failures. However, failures related to malfunctioning technical equipment may sometimes be predicted and avoided.

2.3.1 Maintenance Management

There are three main approaches to maintenance management.

1. **Run-to-Failure Management.** Also known as corrective maintenance. This approach is an "If it ain't broke, don't fix it" management approach. An operator does not spend any money on maintenance until a machine or system fail to operate. However, this is known to be the most costly strategy [Mob04].

2. **Preventive maintenance.** Most Norwegian DSOs lean more toward this approach as they perform basic preventive tasks, such as live inspections, maintenance planning and small adjustments. This approach is sometimes based on statistical characteristics, such as hours of operation and Mean Time To Failure (MTTF)[Mob04].

3. **Predictive maintenance.** PdM involves foreseeing breakdown of a system or component by detecting early signs of failure in order to make maintenance work more proactive, saving money by ensuring a more reliable operation. This is done by utilizing real-time data analytics in combination with historical data to predict problems before they occur and conduct PdM, eliminating costly downtime.

The bathtub curve in Figure 2.7 is often used in reliability engineering. It shows the relative failure rate of an entire population of equipment over time. When the early, random, and wear-out failures are combined, they form a shape resembling the cross-shape of a bathtub, hence the name. The likelihood of failures due to ageing rise slowly in the steady-state phase, but increases remarkably at the wear out part of the curve. The curve often serves as a basis for predicting the Mean Time Between Failure (MTBF) in maintenance management.

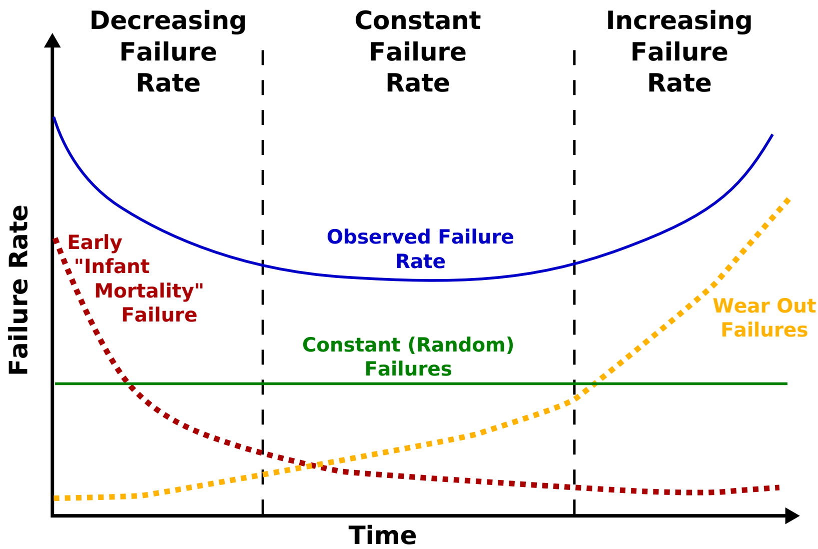

Figure 2.7: The Bathtub Curve [Suh15]

When components operate in different environments, unique circumstances come to play and individual components are affected by various conditions. Therefore, viewing an entire population of components as equal, using calculations of MTBF in order to know roughly when to replace them, is problematic. Such a maintenance strategy does not properly take unique circumstances and conditions into considerations. PdM can be utilized to examine components individually to determine their state. This way, operators can avoid replacing non-failing components. Rather, they might be able to replace components that are near failure, although the age of that component indicates that it "should not" fail at this stage.

Many industries, especially the ones where reliability is crucial, has started to adapt PdM in order to improve reliability, safety, availability and efficiency, as well as to protect the environment.

Grid operators often perform maintenance on equipment when it is already too late (reactive maintenance), or on equipment that does not require maintenance (preventive maintenance) [Con16]. Due to scale and complexity, minor technical malfunctions can result in reduced efficiency and significant financial losses. Malfunctioning equipment in the power grid might also result in outages for consumers.

The deployment of physical sensors, such as Power Quality Analyser (PQA) and Phasor Measurement Unit (PMU), has lead to an increase in available data sources from the power grid. Combining data generated by these sensors with software using machine learning techniques trained on large data-sets, could prove efficient in predicting and giving early warnings on potential faults and instabilities [ATHU18]. A machine learning model trained on the "normal state" might be able to spot outliers in the observations (anomaly detection), which may provide awareness of equipment in need of maintenance. PdM systems may additionally utilize service and repair history of components to help predict MTTF more accurately.

With a successful implementation of PdM, maintenance goes from being primarily reactive to becoming more proactive. It enables more efficient scheduling of service and part replacement ahead of failure, when it has the least impact on operation. Additionally, PdM can contribute in maximizing interval between repairs. A more proactive approach to maintenance is beneficial for both DSOs and consumers, through increased availability, reliability and safety in the power grid.

Numbers from the US Department of Energy [oE10] show that a functional predictive maintenance program can provide up to 10 times return on investment. Maintenance costs can be reduced by up to 30%. Additionally up to 75% of breakdowns may be eliminated, although the level of effectiveness and cost savings varies between industries. As sensors and telecommunication infrastructure is already implemented in the power grid, PdM is a field DSOs should be interested in investigating further.

2.4 Machine Learning

This section gives an introduction to the ideas and methods of Machine Learning (ML) relevant to this thesis. The motive of this section is to provide a basic understanding of the capabilities and challenges related to ML. General ideas will be discussed, before comparing common approaches when ML is applied to problem solving. Finally, we discuss the challenges related to data preprocessing and domain knowledge, and how these may be dealt with.

2.4.1 History and General Idea

During the last 10-20 years, the ubiquity of the Internet has lead to an explosion in the amount of data generated and stored every day. Though ML has seen an upswing in popularity and boost as a buzzword during the last decade, the concept is not new. The fundamental methods and mathematics of ML were proposed more than 60 years ago [For]. Since then, technological advancements have led to reduced costs of computer processing power and increased availability of storage capacity. As a consequence, many industries have invested heavily in research on how their data can be used to generate actionable insight and clinical decision support.

The general idea of ML is that we want to use historical data to make predictions on the future. In mathematical terms, we want to use some set of independent variables (*features*) to predict the value of a dependent variable (*label*). The independent variables of interest compose the *feature-set*, and the dependent variable is the corresponding label. The term *example* is used about one set of features coupled together with the corresponding label of that specific feature-set. Further, the term *dataset* is used about the set of all such examples. When quantity of available data is discussed, we refer to the amount of examples in the dataset.

Features and labels can be either numerical (continuous) or categorical (discrete). A problem where the goal is to predict a numerical label is called a regression problem, while making predictions on a categorical label is called a classification problem.

To further explain the terminology, we use an example problem from the power grid domain. In this regression problem, we want to build a model that is able to predict electricity prices based on weekly measured precipitation and the day of the year (from 1 to 365). We denote the measured precipitation as x_1, the day of the year as x_2, and the corresponding electricity price forecast as y. x_1 and x_2 are the features of the feature-set X, and y is the label.

After collecting a sufficient amount of data, the dataset S contains n examples of the feature-set X coupled together with a corresponding label y, as in Equation 2.1. The dataset S is then used to to train the model. During training, the model is fit to describe how the chosen features maps to the different labels.

$$\begin{aligned} S &= \{X_i, y_i\}_{i=1}^{n}\,, \\ X &= \{x_1, x_2\} \end{aligned} \tag{2.1}$$

The task of creating and tuning these models, to describe the real world in the best way possible, is done through statistical optimization algorithms. This is what we refer to as ML. Some of the more advanced use cases of ML are image recognition, product recommendation systems and fraud detection.

When training ML models, we provide a dataset to the algorithm of our choice, as well as some algorithm-specific parameters that makes sense for the problem and dataset at hand. As more examples from the dataset are provided, the algorithms adjust the model to better fit the examples presented. After having provided the algorithms with a sufficient amount of examples, we let the model predict labels of a previously unseen example.

2.4.2 Artificial Intelligence vs Machine Learning vs Deep Learning

The terms artificial intelligence (AI), machine learning (ML) and deep learning (DL) are often used interchangeably. To dispel confusion, we introduce a simple description of the relationship relationship between the terms. ML is a subfield of AI, while DL again is a subfield of ML, as Figure 2.8 illustrates. AI is a broader, more general term used about bringing intelligent behaviour into machines. ML refers to a more specific application of AI, where machines learn patterns based on observed data, so the way it performs a task improves with experience. DL is a field given much attention lately for its performances at benchmark tasks in text, speech and image processing. The term usually refers to the technique of using multiple layers of Artificial Neural Networks (ANNs) to progressively "learn multiple levels of representation and abstraction that help to make sense of data such as images, sound, and text" [DY14].

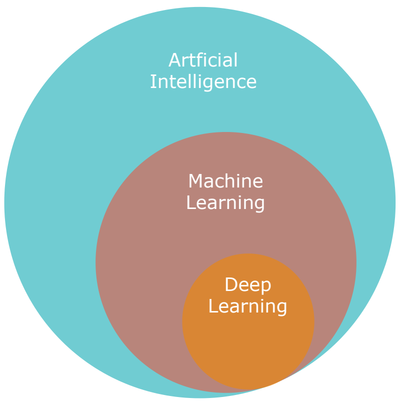

Figure 2.8: Relationship between the fields of AI, ML and DL.

2.4.3 Supervised vs Unsupervised Learning

Supervised Learning

So far we have discussed cases where our dataset consists of feature-sets and its corresponding label. This means that the learning process is done using a ground truth, with prior knowledge of what the output value for a specific feature-set should be. The goal is to learn a function that best approximates the relationship between input and output observed from the data. This is what we call *supervised learning*.

The Random Forest Algorithm

A commonly used algorithm for supervised learning is the Random Forest (RF) algorithm, which can be used for both classification and regression problems. RF uses a technique of building multiple decision trees based on the features of the dataset. For classifications, the model then combines the predictions from these trees into a majority voting system, where the final output class will be the one with the majority of votes, as illustrated in Figure 2.9.

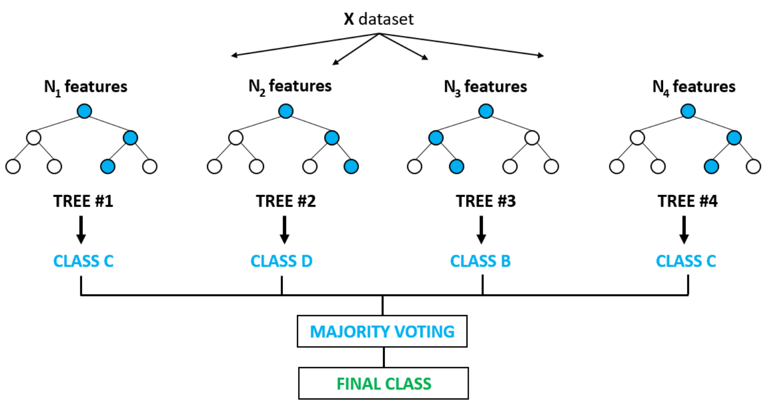

Figure 2.9: Random forest illustrating the majority voting of decision trees.

Unsupervised Learning

ML may also be applied in cases where we have no prior knowledge of the output value for a specific feature-set, meaning the dataset contains no labels. This is called *unsupervised learning*. In this case, the goal is often to find interesting structures in the data and to gain insight which can be further used to produce hypotheses. Clustering algorithms are commonly used for this purpose. As opposed to for supervised learning methods, there is not always a specific way to measure the performance of an unsupervised learning method, as there is no ground truth to compare with. Figure 2.10 illustrates the difference between a supervised classification problem, and an unsupervised clustering problem.

In situations where the dataset contains numerous examples, but we have no knowledge of how to classify these examples, unsupervised learning can be used to provide valuable insight. Sometimes we encounter situations where we have knowledge of what the normal state of a system looks like, but limited or no knowledge about what abnormal states look like. Several real-world problems introduce us to cases like this, where we only have access to information about one of the possible classes. In such cases, the dataset can sometimes be used to detect examples that deviate significantly from the class we have knowledge about. Methods for solving problems like this are called One Class Classification (OCC) algorithms, and are often used for anomaly detection. Anomaly detection is an important tool for industries where detection of abnormal or suspicious behaviour is of interest [CCV08].

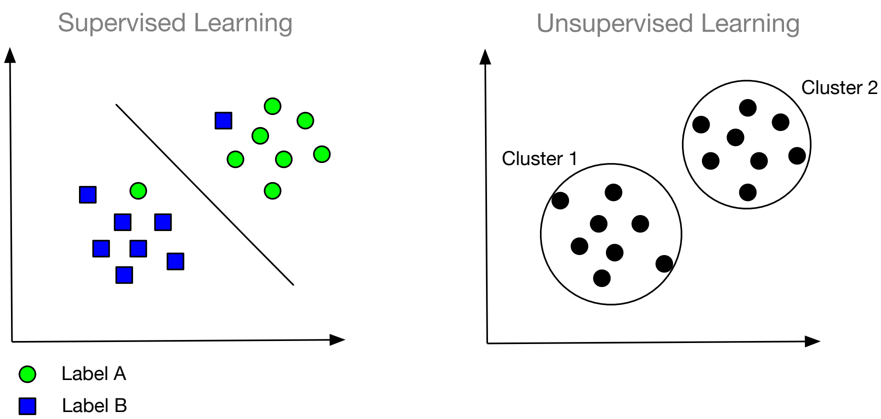

Figure 2.10: Simplified illustration of supervised and unsupervised learning.

One-Class Support Vector Machine

A commonly used OCC algorithm for anomaly detection is the One-Class Support Vector Machine (One-Class SVM) [KM09]. The idea is to use the dataset to find a function which returns either a positive or negative value, depending on how similar the test data is to the examples in the applied training set. The function is made by fitting a hypersphere that includes most of the training data (Usually 90-100%). The amount of examples included in the hypersphere is dictated by the NU parameter, a parameter set by the developers, which should be set based on attributes on the training set.

Figure 2.11 illustrates the fitted hypersphere in a simple classification problem with only two features. Whenever predictions are made on a data point, a scoring-function will output a value describing the distance between that data point and the centre of the hypersphere. All data points which fall outside of this hypersphere will be labelled as an outlier (negative value), while data points inside the hypersphere are labelled as a normal observation [LSKM04]. A data point labeled as an outlier indicates an abnormality.

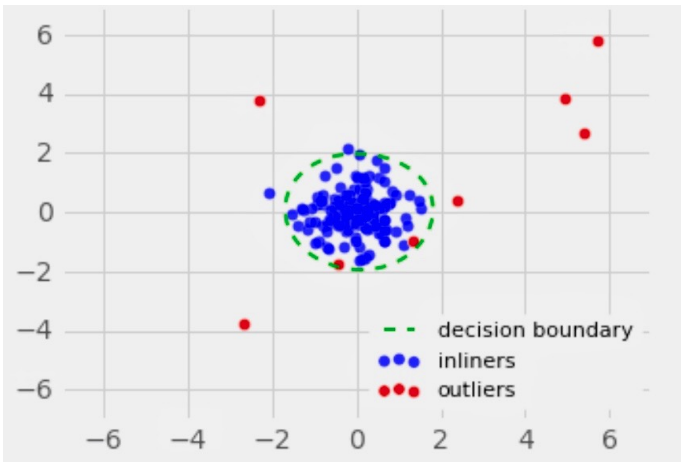

Figure 2.11: Simplified illustration of a one-class SVM.

2.4.4 Data Preprocessing: Providing useful data to the algorithms

Before ML can produce valuable insight and prediction models, the data needs to be thoughtfully handled during the data preprocessing step. The preprocessing step is a significant component of a ML project [RWA+12]. It involves dealing with missing data, scaling values, conversion of categorical features to numerical features, splitting the dataset, as well as deciding which features to use.

Missing Data

Missing some data is a common problem. For different reasons, some examples in the dataset might be missing a value for one or more of its features. For instance, a feature originating from a sensor with limited power access is prone to have some periods of missing data. The best way to handle missing data depends on the context and domain explored. In some cases, it makes sense to assume a missing value can be replaced by the average, or median, of that specific feature from other examples in the dataset. In other cases, it might make sense to drop the entire example from the dataset if a feature-value is missing. If dropping the example removes crucial information, or taking the average does not make sense (such as for categorical features), other methods of handling missing data have to be assessed.

Feature Scaling

Feature scaling is a method to scale numbers of varying magnitudes from different features into the same range. Some of the commonly used ML algorithms, including Support Vector Machines (SVMs) and K-Nearest Neighbors (KNN), use the distance between data points in computations for training a model. As a dataset usually contains features with varying magnitudes, feature scaling is a crucial step in the data preprocessing. While the age of a person will vary from 0 to about 100, the salary of a person might vary between 0 and 1,000,000. Thus, if age and salary make up the feature-set, the magnitudes of the features are highly varying.

Consequently, the age of a person will practically be negligible when computing the distances between data points without feature scaling, as the magnitude of salary, is much higher. There are multiple ways to scale features. A common approach is to scale all features up or down to fit in the range [-1,1] or [0,1]. For ML methods not based on distance calculations, such as decision trees and naive Bayes, feature scaling is not always necessary.

Categorical Data

Categorical features, such as nationality or movie genre, can not be directly compared in the same way as numerical features. Therefore, when using distance-based ML methods, categorical features needs to be transformed and encoded before they can be interpreted by algorithms. A common way of solving this is by using one-hot encoding. In this scheme, we first explore how many different values that exist for a specific categorical feature in our dataset. Then, for every possible value of that categorical feature, we create a new numerical feature which is given the value 0 or 1. This way, a categorical feature is instead represented by multiple numerical features. Table 2.1 and 2.2 illustrates the one-hot encoding scheme.

Table 2.1: Categorical text data

Country
France
Spain
Germany
Spain
Germany
France
Spain
France

Table 2.2: One-hot encoded variables

France	**Germany**	**Spain**
1	0	0
0	0	1
0	1	0
0	0	1
0	1	0
1	0	0
0	0	1
1	0	0

Domain Knowledge

While some correlations and dependencies between variables can be found through data exploration, possessing domain expertise is an advantage in the process of gaining insight from the data. A fundamental understanding of the problem to be solved and the properties of the available dataset, can be time-saving and crucial in order to pick the right features for the ML algorithms to operate correctly. Domain knowledge includes all technical, social and legal factors that may come into play.

Feature Extraction

Feature extraction means decreasing the size of the feature-set. An argument for decreasing the size of the feature-set is to reduce the computational power required for training the model. Additionally, the prediction accuracy of the model may suffer if the feature-set contains redundant features, or features that don't make sense to include for the specific task, as it may confuse the algorithm.

The task of feature extraction is to find a subset of features from the features in the dataset, that maximizes the ability of the learner to create a well-performing model. This process will reduce the dimension of the dataset (reduce the number of features) and remove redundant or irrelevant information, making the remaining feature-set more appropriate for the problem at hand. Deciding what features to include and exclude from the raw dataset is a challenging task, and the performance of the ML model depends on a well-thought selection.

Skewed Datasets

When supervised learning is used, another aspect of feature extraction is the balance of the dataset. In a classification problem with two distinct classes, having a well-balanced dataset would mean having a reasonable amount of examples from both of the two possible output classes. Most of the time when ML is applied to real-world problems, the dataset available will contain some degree of class imbalance. For many of the most used ML algorithms, an imbalanced dataset will often lead to a prediction model biased to predict the dominant class in the training set. Techniques for managing this challenge include undersampling examples from the dominant class or oversampling the underrepresented class. Other, more sophisticated techniques also exist.

Splitting the Data

Finally, before we start to train a model, the dataset needs to be split up into a training set and a test set. The training set is then used for training the model before the test set is used to measure the accuracy of the model. Normally a training set contain 70-80% of the examples in the dataset, the remaining 20-30 % make up the test set. A comparison of the predicted and the true labels in the test set will give an indication of how well the model generalizes to previously unseen cases.

Importance of Data

When the application of ML is considered, a common topic is the availability of sufficient amounts of data for ML to be used. Quantity of data often gets more attention than the actual quality of data, even though high-quality data is a prerequisite for being able to create models that generalize well to new cases. Sometimes data can be transformed into high-quality data through the steps of preprocessing and feature extraction, but this is not necessarily the case. If useful data is not present in the dataset, the mentioned steps will not magically generate high-quality data for solving the problem we want to solve. What we need is a sufficient quantity of high-quality data.

2.4.5 Performance Evaluation

Before any ML model can be put into production and used as an analysis tool, its performance needs to be evaluated. Depending on the method and algorithm used for training the model, different ways to measure the accuracy of the model exists. For a classification problem, a common and natural measure is to count the number of correct and false predictions made on a test set. In case of a binary classification problem, a confusion matrix containing the number of true positives, true negatives, false positives and false negatives is often used. These numbers can also be used for calculating other performance metrics, to better adapt to the domain in which the model is going to be used [Tow].

Cross-validation is a common technique used to ensure that the model does not rely too much on the initially chosen training set. K-fold cross-validation is done by splitting the training data into k folds, then training on k-1 folds before testing its accuracy on the last fold. All of this is done k times. The error of the model is then averaged across the k folds, which gives the cross-validation error.

Overfitting

Overfitting is defined as "the production of an analysis which corresponds too closely or exactly to a particular set of data, and may, therefore, fail to fit additional data or predict future observations reliably" [oxf]. This means that the model will not generalize well to new data. Whenever a ML model achieves significantly better accuracy on the training set than on the test set, chances are high that the model is overfitted. The overfitted model in Figure 2.12 shows a model that learns to fit noise or random fluctuations in the training data. This is learned to such an extent that it negatively impacts the performance of the model.

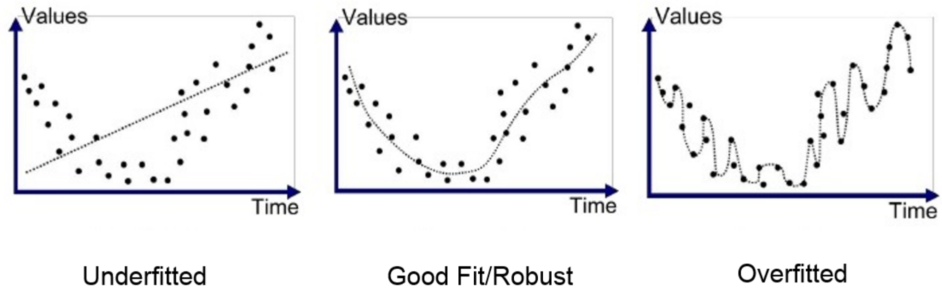

Figure 2.12: Underfitted, good fit, and overfitted machine learning model [Anu18]

2.4.6 Libraries

A library is a tool with a collection of functions and methods which can be used to perform special actions, without writing the code needed to perform the operation. Therefore, using the programming language with the most fitting libraries is beneficial.

Python is considered to be the best language for machine learning [Nau15]. The simple syntax of Python combined with an active developer community has lead to many well documented open-source libraries. Many libraries for different tasks makes ML in python uncomplicated [Har12]. Next, we introduce four important libraries used in our study for to perform ML in python.

NumPy is a mathematical tools library. Functions can be called through the NumPy library to perform advanced mathematical operations. As ML are based on mathematical models, NumPy is one of the most essential libraries for ML. Further, **Matplotlib** is a python library used to plot high-quality graphs. The **Pandas** library is considered the best library to import and manage large datasets in Python [Kir19]. **Scikit-learn** is a ML library built on the aforementioned libraries, and a great tool for performing data preprocessing and ML algorithms.

Chapter 3
Related Work

In this chapter we summarize research related to the scope of this thesis, to provide an overview of the state-of-the-art. We have included papers which have investigated new ideas concerning ML in the operation of the power grid, as well as papers which have studied more specific challenges tied to our approach. Research related to predictive maintenance and forecasting in grid operation are also discussed. Some of our assumptions are based upon the following papers.

According to [ATHU18] monitoring based on measuring instruments in the Norwegian grid has been sparse, and existing research on the power quality signature of fault events and disturbances is limited. This paper is one of our motivating factors for choosing predictive maintenance as our focus for this thesis. In this paper, it is emphasized that possibilities within fault prediction exists in the Norwegian grid based on available sensor data which the various grid operators collect. Their paper suggests that some types of faults are likely to have a signature pattern in advance of the fault happening. Failures related to component faults caused by humidity, salt, ageing, etc. develop over time and is therefore assumed to be possible to detect prior to the actual failure. They conclude that it remains to be established what time resolution and duration is needed for robust detection and prediction. Henceforth, our intention is to extend their research and propose a technical solution to their assumptions.

Already in 2001, Nikolaos Hatziargyriou included a chapter on ML applications to Power Systems in his book on Machine Learning Applications[Hat01]. He points out that forecasting of electricity demand and production volumes are suitable tasks for applying ML. It is also stated that 'real-time measurements of high actuality can detect critical situations and predict failures'. Hence, the research concept conducted in our thesis is not a new idea.

A paper from Texas A&M University researchers in 2009 observed anomalies developing ahead of faults and outages on 60 feeders, to comprehend how those events could be avoided with intelligent monitoring [RBC+09]. The paper documented fault and failure conditions that could reduce feeder reliability and service continuity, and concluded that the number of outages could be significantly reduced by early warning and detection of the faults which later escalated to feeder outages [RBC+09]. Figure 3.1 and 3.2 presents one of many observations made in [RBC+09] prior to an outage.

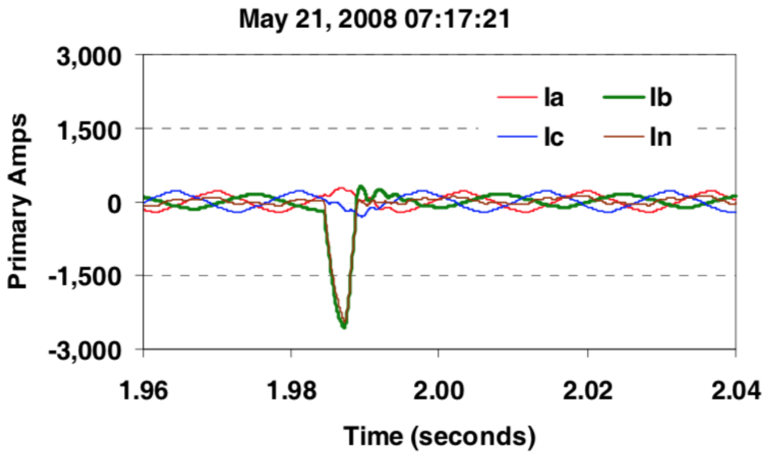

Figure 3.1: Wildlife fault that stressed connector [RBC+09]

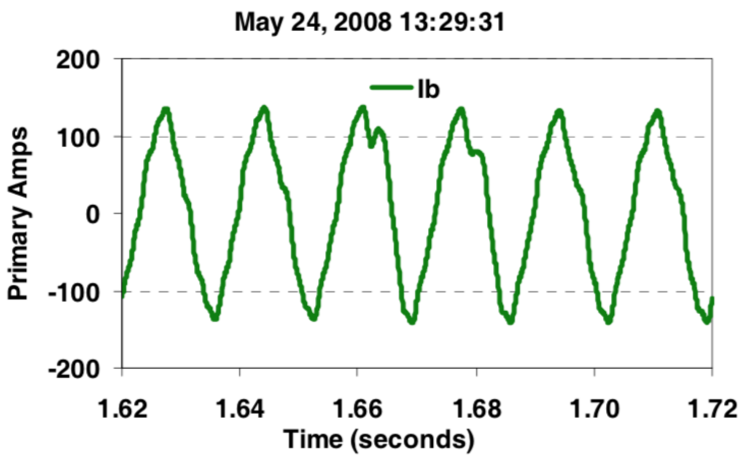

Figure 3.2: Anomalies following wildlife stress, prior to outage [RBC+09]

The wildlife stress on the connector seen in Figure 3.1 caused a fault to the grid, and service was out for an hour. Figure 3.2 shows that three days later the same phase on the same feeder shows an anomaly. The anomaly seen in Figure 3.2 continued intermittently for six days. 'On May 30, 2008, a customer reported lights-out and "arcing and sparking" wires on top of the pole outside his home. The responding crew found that the clamp connecting the primary phase conductor to the customer's service transformer had burned open' [RBC+09]. The observations shown in Figure 3.1 and 3.2 indicates that early warnings are possible with intelligent monitoring. However, measurements done by the instruments in their research was of very high resolution, with millisecond scale, which is not available for us in this thesis' work.

A pioneering project within predictive maintenance in the power grid used historical grid data to create comprehensive models that aim to assist power companies with maintenance planning and decision support, by creating a vulnerability ranking system of components as well as MTBF estimates [RWA+12]. They emphasize the challenges of working with historical grid data not initially meant for predictive purposes. Accordingly, data preprocessing was considered the most challenging step in their research. Further, they underline that power companies often keep historical data records, however, these are not used for predictive purposes.

Instead of using the same variables for predicting grid failures, the NYC research [RWA+12] developed specialized processes for predicting different ranking systems. The data used by their ML models include information about past events, such as failures, repairs and power quality, as well as asset features, such as type of equipment, manufacturer and date of installation. From [RWA+12], we learn the challenge of the preprocessing step when working with such imbalanced datasets, and of managing data from different sources. Nonetheless, their research is an inspiration for proving how studies of this kind may be of help for grid operators.

A project in Portugal proposed a system for predicting events in the Portuguese power grid, using data provided from assets combined with exogenous variables such as weather data [VVM+16]. Their paper included testing of both SVM and RF as classification methods. They suggest using a sliding window approach for labelling the data, then makes use of one RF model and one SVM model as classification methods. Their prediction interval length was set to one day (24 hours). But, their feature window contained features from two days, and a label indicating if an interruption occurred the following day. [VVM+16] Concludes that the performance of SVM and RF classifiers are quite similar.

Target events considered in the study from Portugal were interruptions of power supply, indicating an outage or transformer faults. Their exogenous variables used in their study are the maximum and minimum temperature, maximum gust speed, and most extreme weather event, such as the occurrence of a thunderstorm. Our research intends to utilize similar techniques regarding data labelling, such as the sliding window approach coupled with events data revealing if an interruption occurred during the label window or not. To evaluate their performance they apply two different methods, 10-fold Cross Validation (CV) and a 70% training and 30% testing approach, similar to what we intend to do.

A paper from Politecnico de Bari in Italy presents a data aggregation framework for reducing the size of large datasets while keeping the usability for decision support and predicting grid failures [RLS14]. Their research investigates ways to aggregate data generated by energy distributors in Italy, in order to reduce data volumes. After the volume reduction, the researchers try to predict grid faults. The research point out that the early data cleaning and aggregation steps are critical stages which need strong collaboration between analysts and domain experts. Their cleaned data consists of statistical indexes used to characterize current variations, such as maximum, minimum and average values, as well as variance and standard deviation. These parameters were used to build a decision tree classifier before 10-fold CV is used to measure the accuracy of their classifier. We intend to apply their findings in our study, utilizing ideas from their framework for data aggregation towards data from the Norwegian power grid. Just like the studies from New York [RWA+12] and Portugal [VVM+16], the paper from Politecnico de Bari [RLS14] input both weather and grid data to their machine learning models. The case study presented in [RLS14] is a root cause classification of historical events in the power grid, classifying 77.8% of events correctly.

Table 3.1 contains an overview of some related work on the topic of predictive maintenance in the power grid. The left-most column contains the publication, the articles may be looked up in the bibliography. In the next column, we have outlined what type of data the publication has used for their research. The "method" column contain brief information of how the data has been processed, e.g. what their research investigated or classification methods used. The right-most column, "results" is a short explanation of the findings and conclusions of the corresponding paper.

Publication	Data source	Method	Results
[RBC+09]	60 feeders with documented faults and outages.	Examines if events could be avoided if intelligent monitoring was in place by reviewing pre-fault data.	Concludes that reliability can be significantly improved with condition-based maintenance.
[RWA+12]	Past events, such as failures, replacements, repairs. Also, asset features, such as the type of equipment, environmental conditions, manufacturer, components connected, date of installation.	Uses supervised bipartite ranking algorithms to predict the risk of failures for components, with time-shifted features/labels (up to a year). Ranks for various types of equipment, i.e., 1000+ high voltage feeders.	60% of failures occur in the 15% of feeders ranked as most susceptible to failure. Concludes that data already collected by power companies can be harnessed to predict and assist in preventing grid failures.
[GKWK15]	Post-fault recovery voltage measurements in different operating conditions.	A support vector machine classifier to predict the stability status of the system.	
[RLS14]	27 months of observations and 2438 interruptions. Aggregated current data, line data and weather data.	Aggregated data and a sliding window of 30 minutes. Predicting possible grid failures with decision tree classifier.	Correctly classifies 77.8% of the events.
[VVM+16]	7 months of operation data generated by 300 transformers and weather data. Out of 60,000 labels, 277 interruptions are found in the dataset.	Sliding window classification approach, bag-of-words event representation, and tests random forest models and support vector machines.	Up to 0.75 area under the receiver operating characteristic curve in testing.

Publication	**Data source**	**Method**	**Results**
[ATHU18]	Data from FASIT together with graphs of grid data from the Norwegian power grid.	Presents the idea of performing monitoring and fault prediction methods on the Norwegian power grid.	Given the right data, there should be opportunities for developing a machine learning based fault identification and prediction algorithm.

Table 3.1: Table summarizing various related articles

Some of the papers, such as [RBC+09] and [ATHU18] purely identifies possibilities with creating actual models. The NY project [RWA+12] created ranked lists of feeder, components and manhole events. As they created different models for each type of event their results vary, however, their overall conclusion is that all events are worthwhile to model for prediction. While [GKWK15] uses post-faults voltage to predict stability status of the network.

[RLS14] creates a model for classification of faults, based on current measurements and weather data. We intend to have an approach more similar to [VVM+16], and investigate if the prediction of faults within a given time-window is possible, based on grid data from a Norwegian DSO in combination with weather data. Such a project has, to our knowledge, not been performed on the Norwegian power grid before.

Chapter 4
Technical Approach

This chapter provides information on the process of data collection, preprocessing, and the procedures to create a machine learning model. Throughout this chapter, we explain why we chose particular procedures. The chapter may be considered a guideline for building similar models.

Figure 4.1: Visualisation of the main tasks included in the technical approach.

Method selection and implementation depend on the characteristics of the data collected. The technical approach followed in this chapter may be visualized as the process in Figure 4.1. Below we describe the process from data retrieval to implementation of the ML algorithms.

4.1 Data Collection

In this section, we review the data collection task. At the beginning of this project, we had limited knowledge about the data available. Also, we did not know from which sources we could expect to receive data. In order to collect data, we contacted several DSOs. The goal of contacting several DSOs was that at least one of them possessed grid data which they could share with us. Through a review of the partners in the Centre for Intelligent Electricity Distribution (CINELDI) project [cin], and exploration of pilot projects these partners were a part of, we decided upon a few DSOs to contact.

The DSO we received data from has an ongoing pilot project in approximately 30 of their substations. In these substations, the DSO has installed several sensors for monitoring equipment. Furthermore, this DSO spends more than 10 million NOK a year on CENS for non-planned outages, implying faults should exist in their data. However, only two of the substations equipped with sensors recorded a failure during the last 12 months. The data from these substations were relevant for us to inspect closer. Also, fault logs from their SCADA system would be relevant.

After valuable conversations, through semi-structured interviews, with two DSOs, we signed a confidentiality agreement with one on these. We received data from substations *S1* and *S2* for two specific dates which contained faults. The received data contained 24 hours of measurements with a resolution of one minute. We requested more data from other periods in *S1* and *S2*, as more faults and longer intervals of continuous measurements during the substations normal condition could be beneficial. Information about the root causes of the failures in the datasets already received were also requested.

Unfortunately, the DSO experienced technical difficulties when trying to export more data from *S1* and *S2* from their SCADA-systems. Eventually, we received a dataset from a substation *S3*, containing one year of hourly aggregated measurements. The DSO helped explain the technical details in the data we received.

For components in the power grid, it is reasonable to believe that some exogenous variables, like weather data, might affect the system state. While rainfall and snow might affect underground power cables, wind can cause disturbances through falling trees and vegetation. Additionally, outside temperatures affect electricity consumption, and thus also the load on the power grid. Therefore we have included data on wind and rainfall collected from weather stations close to the substations in this study.

Weather data is freely available on JSON-format through a RESTful API provided by the Norwegian Meteorological Institute. The API gives access to "quality controlled daily, monthly, and yearly measurements of temperature, precipitation, and wind data" [meta]. To access the API, we created a user to get the credentials needed for logging in to the service. Through the API documentation, we found guidelines on how to access information about weather stations, and how to extract data from these sources through HTTP-requests with Python [metb].

We located the weather stations which were physically as close as possible to the substations of interest, and inspected available data from these weather stations. Not all of these weather stations contained data on both wind and precipitation from the time intervals of interest. As a result, we had to retrieve measurements of precipitation from one weather station, and measurements of wind from another.

4.2 Data Characteristics

We investigate data characteristics to find useful traits for our purpose. Data characteristics are significant in the phase of algorithm selection. The purpose and importance of different measurements in the dataset should also be understood. Due to limited domain knowledge, this too is part of investigating the data characteristics. This section presents the most significant characteristics discovered during the inspection of the data.

In total, we received data containing sensor measurements from three substations. From one of the substations, *S3*, we received hourly aggregated values from the year 2018. However, no faults occurred in *S3* during 2018. For substations *S1* and *S2*, we received measurements, as well as event logs. The data covered two specific dates, October 27th, 2018 and January 1st, 2019. *S1* and *S2* experienced failures resulting in outages on both of these dates. Table 4.1 summarizes the characteristics of the received datasets.

	S1	***S2***	***S3***	**Shared**
Data Interval	27.10.2018 & 01.01.2018	27.10.2018 & 01.01.2018	09.01.2018 - 31.12.2018	-
Resolution	Every minute	Every minute	Every hour	-
Event logs included	Yes	Yes	No	-
Faulty days	2	2	0	-
Sensors measuring voltage	30	24	36	24
Sensors measuring current	15	12	18	6
Sensors measuring power	5	4	6	4
Sensors measuring volt-ampere	12	9	14	8

Table 4.1: Overview of the data collected from substations

Table 4.1 shows similarities between the dataset from substations *S1* and *S2* especially considering the data interval, resolution and faulty days. Data from substation *S3* differs significantly from *S1* and *S2*. Further, the number of sensors measuring voltage, current, and power varies for each substation. Additionally, the feeders of *S1* and *S2* have a voltage of 22 kV, while the feeders of *S3* have a voltage of 11 kV.

The papers described in the Chapter on related work, varies between having 277 to 2438 faults in their datasets. Meanwhile, the datasets we have received contain only eight failures in total. Table 4.3 shows the times when failures occurred in *S1* and *S2*. Having such few examples and few failures in our dataset may bring more uncertainty to our results.

Upon further inspection of each of the measurements, we observed some measured values were zero through the entire dataset. These zero-values indicate inoperative sensors. Also, several sensors produced long sequences of repeating values, even though they stem from variables of continuous nature. Some of these sequences contained repeating values for up to three minutes. After inquiring with the DSO about this, the explanation was that the SCADA-system does not register a change in measured value until the change is more significant than a certain threshold.

Some measurements were comparable across the substations. A total of 24 voltage sensors and six current sensors appears to be measuring similar physical components across the three substations. In the same way, the number of comparable measurements appears to be four for power and eight for volt-ampere. However, all measurements on power and volt-ampere in *S1* and *S2* appeared to differ significantly from the corresponding measurements in *S3*, both in amplitude and variance. Furthermore, in the dataset from *S3*, all values freeze in the period from 25.05.2018 to 17.08.2018. Consequently, we excluded this period from the dataset.

The characteristics of the data received from the three substations vary in both resolution and duration. To summarize, the dataset from *S3* contains data for a longer time period, while *S1* and *S2* contain measurements from two single days at a time. The measurements from *S1* and *S2* are from periods with registered faults and outages, while *S3* is in normal condition during the entire period.

4.3 Selection of Machine Learning method

Before examining the data received, we planned to follow the approach taken in many of the papers presented in 3. In the experiments of these papers, the data used contained numerous faults and longer periods of measurements. In such situations, supervised learning methods have proved to be efficient for building prediction models.

However, the data evaluation of our data revealed that the datasets from *S1* and *S2* contained short periods of faulty condition data, while *S3* contained a longer period of only normal condition data. Therefore, we decided to reconsider our approach. In 2.4, we discussed some use cases of unsupervised learning methods, including detection of outliers, or anomalous data points. As our data, especially from *S3*, contain long periods with numerous examples of the same class, namely the normal condition of the substation, we decided to explore unsupervised learning as well.

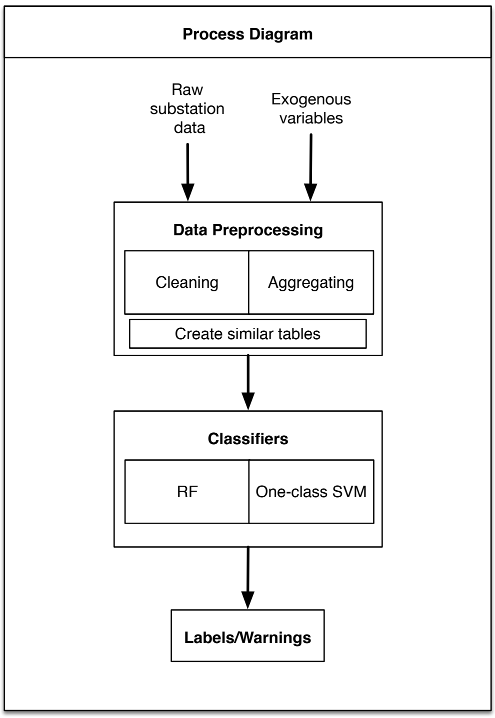

Figure 4.2: Process diagram.

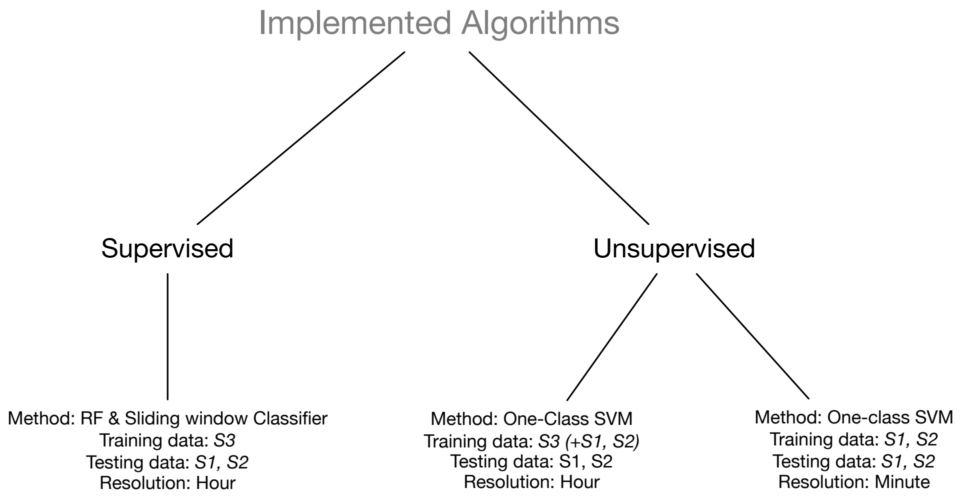

Figure 4.3: The three algorithm approaches conducted, one supervised and two unsupervised.

4.4 The Supervised Implementation

The first implemented approach was a supervised learning approach. With this approach, we wished to train a model on data from one substation over an extended period. The model was designed to recognize recurring patterns before failure. However, due to the limited datasets received, we chose to blend the data from *S1* and *S2* multiple times into the *S3* dataset. We did this to demonstrate how we would implement a ML algorithm on a substation with a sufficient amount of failures during a measurement period. Therefore, the presented supervised approach provides more of a framework for future similar work on how to implement such an algorithm, rather than being an attempt to achieve strong results.

4.4.1 Data Preprocessing

As mentioned in the background and in related work, the preprocessing is often considered the most challenging step of a ML project. The preprocessing steps of the supervised approach consisted of the following steps:

1. Import the dataset.
2. Make datasets from *S1* and *S2* resemble the dataset from *S3* by extracting the similar relevant features into tables, and aggregate *S1* and *S2* to hourly measurements.
3. Blending the datasets from *S1* and *S2* into *S3*.
4. Retrieving the exogenous variables and adding them to the table obtained from (3).
5. Handle missing data.
6. Add labels to the data.
7. Perform feature scaling.
8. Split the table into a training set and a test set.

First we converted the Excel files to CSV format and imported them to Spyder (IDE). Next, we extracted the power and voltage measurements similar in *S1, S2* and *S3*. These were 6 Low Voltage (LV) (400V) measurements and 18 High Voltage (HV) (22kV) measurements, as well as 3 LV and 3 HV electric current measurements.

Thereafter, we aggregated the minute values from the datasets of *S1* and *S2* into hourly aggregated minimum and maximum values, in order to make the tables of *S1* and *S2* similar to that of *S3*. We then mixed data from *S1* and *S2* into the table with measurements from *S3*, in order to make the *S3* dataset resemble a substation with multiple failures throughout a longer measurement period. In total, we mixed ten faults into the dataset from *S3*. The *S3* table now contains several months of hourly minimum and maximum values, including a total of 10 faults. Next, we added the obtained weather variables. We replaced missing data points in the weather data with the mean of the respective column.

In order to train the supervised model, we need to add labels to the data. We utilize the sliding window approach for adding labels to the data based on whether a fault occurs during the next hour or not. We label a row as "1" if failure happens during the next hour, and "0" if there is no fault during the next hour, as illustrated in Figure 4.4.

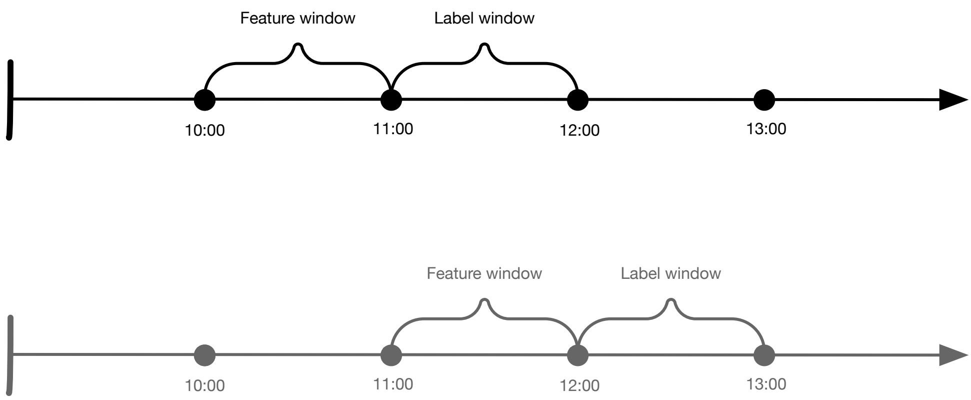

Figure 4.4: Sliding Window

Then we feature scaled the data, which is essentially normalizing the data within a particular range. The standard score is calculated according to Equation 4.1. Where x is the scaled sample, u is the mean of the training samples, and s is the standard deviation of the training samples. The main reason for doing this is to avoid the features with large number ranges to dominate those with smaller ranges [BLB+13].

$$z = \frac{x - u}{s} \tag{4.1}$$

Lastly, we divided the table into two non-overlapping sets, the training set, and the test set. We chose to split the data on November 1st, so the training set contains data from January 9th to November, and the test set contains data from November and December. The training set is then used to build the model before the test set is used to make predictions for evaluating the accuracy of the model. In total, 70% of the data is used for training and 30% for testing (after removing the dates when the measurements freeze).

4.4.2 Predictions

The model has now been trained on historical data to learn how to recognize patterns (if there are any) occurring before a grid failure, and use this data in order to try and predict whether a fault will occur during the next hour or not. As mentioned in 2.4.3 RF classification is used for making these predictions. The results of these predictions are shown and discussed in chapter 5 Results and Discussion.

4.5 The Unsupervised Implementation

The unsupervised approaches are different from the supervised approach. A distinct difference is that the unsupervised procedures did not include labeling of the data before training the model. We implemented two unsupervised approaches with the data at hand. In the first unsupervised approach, we make use of the *S3* dataset, containing no faults, for training the model. The second unsupervised approach takes individual differences between substations into consideration.

We trained a model using One-Class SVM to learn typical patterns from healthy substations, to later be able to spot measurements deviating from the normal condition. The idea of One-Class SVM is described in the section on One-class SVM. In the first unsupervised approach, we train a model using samples containing hourly aggregated minimum and maximum values from *S1, S2*, and *S3*, as well as weather data. For the second unsupervised approach, we train separate models for *S1* and *S2* using minute measurements from only the substation itself. The rest of this section will further describe and clarify the two executed unsupervised approaches.

4.5.1 Data Preprocessing for Unsupervised Learning

First Unsupervised Implementation

During the preprocessing phase of the first unsupervised approach, we created the training set with current and voltage measurements from the *S3* dataset and added wind and precipitation data to this table. Next, we created four tables, two from 27.10.2018(*S1* and *S2*) and two from 01.01.2019(*S1* and *S2*), this made up our test sets. The received datasets from *S1* and *S2* contained minute measurements of both voltage and current from days with grid failures. We aggregated the minute measurements to hourly minimum and maximum values and added wind and precipitation data in order to make the test sets resemble the training set from *S3*.

Table 4.2: Training set and test set for the first unsupervised approach.

Training set	*S3*
Test set	*S1 (27.10 & 01.01), S2(27.10 & 01.01)*

Table 4.2 clarifies that the first unsupervised model uses the large healthy dataset from substation *S3* for training, while the 24-hour datasets provided by *S1* and *S2* were used for testing. After scaling the features in a similar way as in the supervised approach, the tables are ready for training and testing the model.

Second Unsupervised Implementation

In the second unsupervised approach, we utilize the datasets with minute resolution from *S1* and *S2*, to investigate the effect of training and testing on shorter intervals. Additionally, in this approach, we only test a model on data from the same substation as the model was trained on, as illustrated in Figure 4.5 and Figure 4.6. In the two previous approaches, individual differences between substations might negatively affect the models created. However, as each of the models created in the second unsupervised approach only deals with data from one specific substation, this effect from individual differences is removed.

The second unsupervised approach used the same features as the first unsupervised implementation, namely, current, voltage, and temperature. However, the tables in the second unsupervised approach do not include wind and precipitation, as we were unable to retrieve weather data at such a high degree of granularity.

Splitting the Data in the Second Unsupervised Approach

In the second unsupervised approach, we split the data into training and test sets based on the time of failure in the respective datasets from *S1* and *S2*.

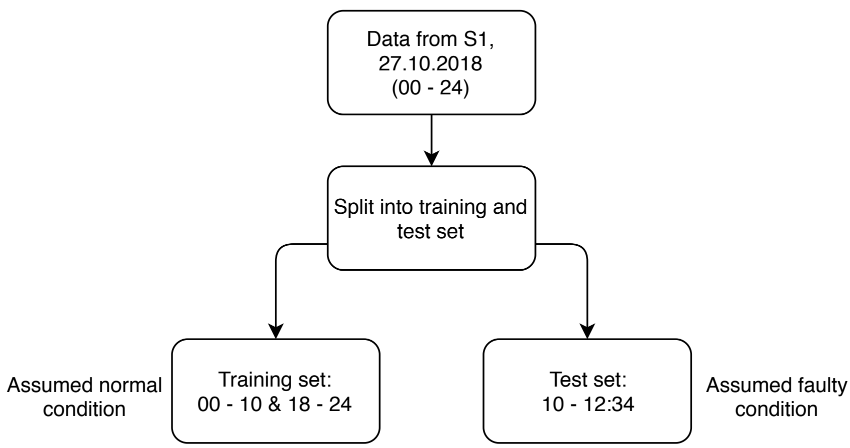

Figure 4.5: Example of training and testing data used in second unsupervised approach.

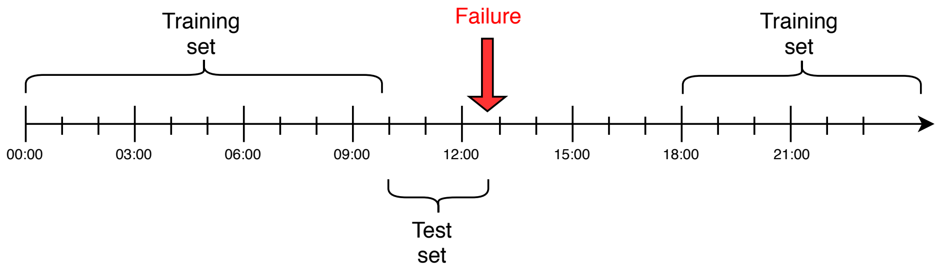

Figure 4.6: The timeline shows the intervals used for training sets and test set with respect to the time of failure, in the second unsupervised approach. The time of failure is not included in the test set. The example corresponds to data from *S1* on 27.10.2018.

The timeline in Figure 4.6 illustrates the intervals used for training and testing. The training set contains measurements up to three hours before failure, as well as data from a period starting a few hours after the fault. We inserted a delay between the failure and the second part of the training set, as we assume measurements from this period might not reflect the normal state of the substation. The test set contains measurements from the period between approximately three hours prior to the failure, up until the point of failure. Table 4.3 gives an overview of when failures happened in the various datasets, as well as the periods used for training and testing.

Table 4.3: Failure times, training sets, and testing sets used in the second unsupervised approach.

	***S1* (27.10.18)**	***S2* (27.10.18)**	***S1* (01.01.19)**	***S2* (01.01.19)**
Faults	12:34 - 12:40	12:34 - 12:40	10:33 - 10:35, 11:01 - 12:07, 14:50 - 14:51	10:33 - 10:35, 11:01 - 12:07, 14:50 - 14:51
Training set	00:00 - 09:59, 18:00 - 23:59	00:00 - 07:59, 15:00 - 23:59	00:00 - 03:29, 18:00 - 23:59	00:00 - 07:59, 16:00 - 23:59
Test set	10:00 - 17:59	09:00 - 14:59	08:00 - 15:59	08:00 - 15:59

The faults in Table 4.3 are obtained from the DSO's event logs. From manual inspection, we observed that some of the datasets had fluctuations which could indicate abnormal behavior. However, this behavior is not reported in the event logs. As we did not want to include these fluctuations in the training sets, we have used slightly different lengths of training and testing intervals. As a result, the largest test set contains 33.3% of its total dataset, and the smallest test set contains 25% of its total dataset.

4.5.2 Parameters

After the tables included identical features, we scaled the features according to Equation 4.1. Next, we chose the parameters for the One-Class SVM, before training these unsupervised classifiers. Identical parameter values were chosen for both unsupervised classifiers. The parameters passed into the unsupervised classifiers are noted in Table 4.4.

Table 4.4: Parameters used for the One-Class SVM

One-Class SVM Parameters	Value
NU	0.05
Cache Size	200
Degree	3
Kernel projection parameter (coef0)	0.0
Gamma	Scale
Kernel	RBF
Maximum iterations	-1
Random state	None
Shrinking	True
Tolerance	0.001
Verbose	False

Most of the parameters in Table 4.4 are default values from the Scikit-learn library. The ones worth noting are NU, Gamma, Kernel, and maximum iterations. The aforementioned NU parameter, discussed in Section 2.4.3, is the upper bound on the numbers of errors in the training example. As we sat the NU parameter to 0.05, at most 5% of the training examples are not included in the constructed hypersphere used for classification.

Our gamma parameter is set to "Scale", which means it uses Equation 4.2 for calculating gamma. Thus, gamma is dependent on the number of features and the variance between the set of samples (X), from the training set. Figure 4.7 shows a simplified illustration of changing the gamma parameter. As illustrated in Figure 4.7 a common effect of increasing gamma is overfitting.

$$Gamma = \frac{1}{N_{features} \cdot variance_X} \tag{4.2}$$

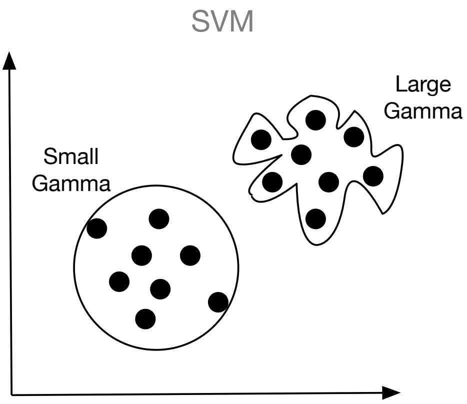

Figure 4.7: The difference between a large and small gamma parameter

Kernel functions are algorithms used for pattern analysis. Our kernel is set to the radial basis function (RBF), which is the most commonly used kernel function in SVM classification. The maximum iterations default value is -1, implying that there is no limit. As the running time still is short in our cases (>1s), we kept the maximum number of iterations to the default value of -1. All the One-Class SVM parameters were kept the same for both unsupervised procedures.

4.5.3 Training the Models

First Unsupervised Model

With parameters set, features chosen, and observations scaled, we can train the model. In the first unsupervised approach, data from *S3* are used to teach the model how a healthy substation behave. We trained the model five separate times, each time using a sequentially chosen 80% of the dataset from *S3*. We did this to ensure that the quality of the data from *S3* was consistent and stable. A high variance between the five models would indicate that the dataset from *S3*, which is assumed to represent a healthy substation, contained periods of non-consistent or unstable data. After having trained the classifiers, we tested each of the five classifiers using the faulty datasets from *S1* and *S2*.

Second Unsupervised Model

In the second unsupervised model, we trained a total of four models. The training set from *S1* on 27.10.2018 was used to train a model for making predictions on the test set from the same date in the same substation. Figure 4.5 and Figure 4.6 illustrates the selection of training and test set. Using the same method of selecting training and test set, we also trained a model for *S1* on 01.01.2019, a model for *S2* on 27.10.2018 and a model for *S2* on 01.01.2019.

The next Chapter, Results and Discussion, provides the results from all models which were trained in the supervised and unsupervised implementations.

4.6 Performance Metrics of Prediction Models

It is common to evaluate the accuracy of a classification model by inspecting the confusion matrix produced by predictions on the test set. Figure 4.8 shows the layout of a confusion matrix. A classification falling into the square denoted TP or TN indicates a correctly classified observation. On the opposite, an observation falls into the square of FP or FN if wrongfully classified.

		Actual Value	
		Positive	Negative
Predicted Value	Positive	**TP** True positive	**FP** False Positive
	Negative	**FN** False Negative	**TN** True Negative

Figure 4.8: Confusion matrix used for visualising model performance

With knowledge about the actual class of an observation, evaluation of a prediction is trivial. By comparing the predictions to the corresponding actual class, one can easily calculate a confusion matrix. In the supervised model, an observation is identified as faulty if the model expects a fault to occur within the next hour. On the occasion that an observation is identified as faulty by the model, and failure did happen within the next hour, the prediction is categorized as a true positive. On the other hand, if a failure did not happen during the next hour, the prediction is categorized as a false positive. Conversely, if the model predicts a failure will not take place during the next hour, but failure did occur, the prediction is categorized as a false negative. A prediction is categorized as a true negative when the model rightfully predicts that failure will not happen within the next hour.

However, in the unsupervised approaches, to decide whether the detection of an anomaly should be considered correct or not, is more problematic. The models trained in the unsupervised approaches are not able to specify the expected time of a failure, as these models are only trained to detect anomalies. Rather, a detected anomaly may imply that the substation is in a state where the probability of failure is higher than usual.

How long in advance indications of a failure can be detected, is dependent on the failures root cause. As we are unaware of the root causes, we are unable to manually label the condition corresponding to an observation as normal or faulty. Hence, there is no actual class describing the condition of the substation at the given time of the observation. Therefore, we have chosen two alternative methods for presenting the results from the unsupervised approaches.

The first method is a simple presentation of the number of anomalies detected during the hours before failure. In the second method, we calculate the statistical significance of the number of anomalies detected, based on the chosen NU parameter. Following is a more detailed description of the two methods.

1. In the first table, we present the number of observations classified in each of the test sets, along with the number of anomalies detected. In the first unsupervised approach, the resolution of the observations is one hour. In the second unsupervised approach, the resolution is one minute. Consequently, the total number of classifications is higher in the second unsupervised approach than in the first.
2. In the second table, we present the results of statistical hypothesis testing. Here, we first determine if the number of anomalies detected before a failure is significantly higher than the number expected under normal condition. For each of the failures, we then check whether a statistically significant number of anomalies were detected before the failure.

The hypothesis statement is that a larger fraction of observations will be classified as anomalies before failure than during a healthy period. To test this, we use the number of anomalies detected by the model on the test set. We test the hypothesis by calculating the probability of obtaining the respective number of anomalies during a healthy period.

The null hypothesis is that the probability of observing an anomaly is 5% (as the chosen NU parameter is 0.05). The alternative hypothesis is that the models will classify more than 5% of the observations before a failure as anomalies. The hypothesis statement presented in Equation (4.3).
H_0: 5% of the observations prior to a failure is classified as anomalies.
H_1: More than 5% of the observations prior to a failure is classified as anomalies.

Mathematically, H_0 and H_1 can be expressed as

$$\begin{aligned} H_0 &: p = 0.05 \\ H_1 &: p > 0.05 \end{aligned} \tag{4.3}$$

For calculating whether to accept or reject the null hypothesis, we define a binomial distribution as a function of the *# anomalies detected.* This distribution defines the random variable X. The number of trials n is the total number of observations prior to a failure, and p is the probability of classifying a healthy observation as an anomaly. The binomial random variable is presented in Equation (4.4).

$$X \sim B(n, p),\ n \in \mathbb{N},\ p = 0.05 \tag{4.4}$$

In the second unsupervised approach, the number of classifications (n) is often large (> 50) and the value of p is small (< 0.2). Consequently, we can use a Poisson approximation for representing the distribution of X, shown in Equation (4.5). The expected number of observed anomalies in a test set is denoted λ.

$$X \sim Poisson(\lambda) \tag{4.5}$$

The *p-value* is defined as the probability of finding the observed, or more extreme, results when the null hypothesis (H_0) is true [p-v]. In our case, the *p-value* is the probability of finding an equal or greater number of anomalies during a healthy period, than the number detected in the test set. The *p-value* is given by Equation (4.6).

$$P\text{-}value = P(X \geq x) \tag{4.6}$$

The significance level is the probability of wrongfully rejecting the null hypothesis. We have chosen a significance level (α) of 1%. If the *p-value* is less than the significance level, we reject the null hypothesis (H_0).

$$P\text{-}value \leq \alpha \tag{4.7}$$

Equation (4.7), will be used for determining if the number of anomalies detected prior to failure is statistically significant. Further, we will determine if all classifications prior to all failures in the test sets are statistically significant to reject H_0.

This chapter presented the procedures from the data collection phase to implementing and evaluating three machine learning models. The models try to predict failures in the grid, based on data collected from a major Norwegian DSO. In the next chapter, we will present the results obtained from these implementations.

Chapter 5

Results and Discussion

This chapter presents the results obtained from the technical approach, followed by a discussion of the procedure, the substance of the results, and challenges encountered.

5.1 Results

This section presents the results in the order of which we conducted the implementation. Hence the results from the supervised implementation are presented first. Followed by the results from the two unsupervised implementations.

5.1.1 Supervised results

During the implementation of the supervised ML algorithm, we blended data from the faults in *S1* and *S2* multiple times into the *S3* dataset, which contained almost a full year of data. This approach resulted in a model tested on the same data of which it has been trained. Training a model on the same faults multiple times causes the model to generalize poorly, the outcome is an overfitted model. The motivation for creating such a model was to develop a framework and a methodology for performing supervised learning for future work possessing sufficient amounts of data.

The confusion matrix from the supervised approach seen in Figure 5.1 below, illustrates the results of an overfitted model. As the same faults occurred in the training and test set, the predictions become unrealistically good and the results from the supervised learning predictions unusable. However, the results demonstrate such an algorithms ability to learn patterns.

		Actual Value	
		Positive	Negative
Predicted Value	Positive	**14**	**0**
	Negative	**0**	**1448**

Figure 5.1: The confusion matrix for the supervised two-class classification.

5.1.2 Results from Unsupervised Implementations

This section present results obtained from the unsupervised approach described in the previous chapter. The distributions and predictions presented in Figures 5.2 - 5.4, as well as Tables 5.1 and 5.2 are based on the same prediction model, namely the one referred to as the "First Unsupervised Implementation". Results presented in Figure 5.5 and Tables 5.3, 5.4 and 5.5 are based on the prediction model referred to as the "Second Unsupervised Implementation".

First Unsupervised Implementation

After using One-Class SVM for training the model, the scoring function described in Section 2.4.3 is used to decide whether or not an observation is classified as an anomaly. If the scoring function, for a given observation, outputs an absolute value larger than a given threshold (decided by the NU-value), the observation is an outlier and classified as an anomaly.

Figure 5.2 shows the distributions of the scoring-functions for the training, and test data. The text below each plot specifies the test set in the corresponding plot. The training data is the same for all four plots. The figure illustrates how the different test sets, compares to the training set, which only contains observations during the normal state. The blue shaded curve is the distribution of the scoring function of the training set, the orange curves corresponds to the scoring functions of the test sets. The vertical red lines indicate the decision boundaries used for classifying an observation as either an anomaly or as a healthy observation. After manual review of the data used in Figure 5.2, we learned that the small distinct peaks observed on the left side of the test data represent the actual faults from the test sets.

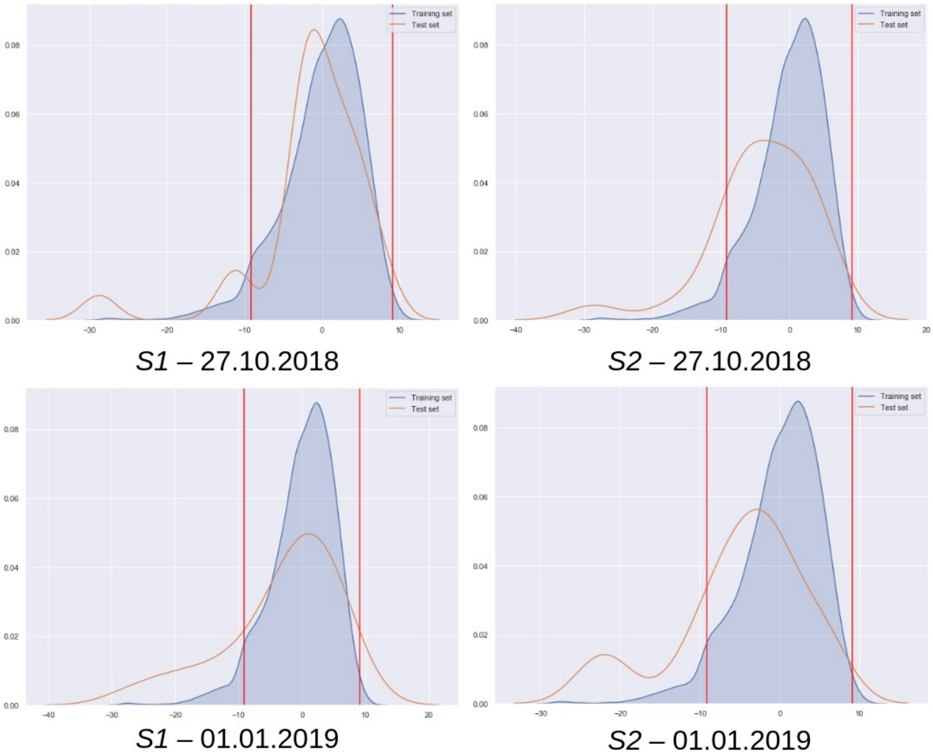

Figure 5.2: Distribution comparison plot with data for all four test sets.

Figure 5.3 and 5.4, presents the models predictions on the test sets from 27.10.2018 and 01.01.2019 compared to the true labels of these observations. As previously described, an observations true label is set based on whether or not a fault occurred during the given hour.

In Figure 5.3, the negative "True Label" observed at 12:00 in left column of the figure indicate faults in both S1 and S2 sometime between 12:00 and 13:00. On the test set from S1, the model detects an anomaly between 11:00 and 12:00, an hour before the actual fault. On the test set from S2, the model detects anomalies between 04:00 and 06:00.

Index	True Label	Predicted Label
2018-10-27 00:00:00	1	1
2018-10-27 01:00:00	1	1
2018-10-27 02:00:00	1	1
2018-10-27 03:00:00	1	1
2018-10-27 04:00:00	1	1
2018-10-27 05:00:00	1	1
2018-10-27 06:00:00	1	1
2018-10-27 07:00:00	1	1
2018-10-27 08:00:00	1	1
2018-10-27 09:00:00	1	1
2018-10-27 10:00:00	1	1
2018-10-27 11:00:00	1	-1
2018-10-27 12:00:00	-1	-1

S1 – 27.10.2018

Index	True Label	Predicted Label
2018-10-27 00:00:00	1	1
2018-10-27 01:00:00	1	1
2018-10-27 02:00:00	1	1
2018-10-27 03:00:00	1	1
2018-10-27 04:00:00	1	-1
2018-10-27 05:00:00	1	-1
2018-10-27 06:00:00	1	1
2018-10-27 07:00:00	1	1
2018-10-27 08:00:00	1	1
2018-10-27 09:00:00	1	1
2018-10-27 10:00:00	1	1
2018-10-27 11:00:00	1	1
2018-10-27 12:00:00	-1	-1

S2 – 27.10.2018

Figure 5.3: Predictions for *S1* and *S2*, 27.10.2018.

Index	True Label	Predicted Label
2019-01-01 00:00:00	1	1
2019-01-01 01:00:00	1	1
2019-01-01 02:00:00	1	1
2019-01-01 03:00:00	1	1
2019-01-01 04:00:00	1	1
2019-01-01 05:00:00	1	1
2019-01-01 06:00:00	1	1
2019-01-01 07:00:00	1	1
2019-01-01 08:00:00	1	1
2019-01-01 09:00:00	1	1
2019-01-01 10:00:00	-1	-1
2019-01-01 11:00:00	-1	-1
2019-01-01 12:00:00	-1	-1
2019-01-01 13:00:00	1	1
2019-01-01 14:00:00	-1	-1

S1 - 01.01.2019

Index	True Label	Predicted Label
2019-01-01 00:00:00	1	1
2019-01-01 01:00:00	1	1
2019-01-01 02:00:00	1	1
2019-01-01 03:00:00	1	1
2019-01-01 04:00:00	1	1
2019-01-01 05:00:00	1	1
2019-01-01 06:00:00	1	1
2019-01-01 07:00:00	1	1
2019-01-01 08:00:00	1	1
2019-01-01 09:00:00	1	1
2019-01-01 10:00:00	-1	-1
2019-01-01 11:00:00	-1	-1
2019-01-01 12:00:00	-1	-1
2019-01-01 13:00:00	1	1
2019-01-01 14:00:00	-1	-1

S2 - 01.01.2019

Figure 5.4: Predictions for *S1* and *S2*, 01.01.2019.

Table 5.1: The table presents the results for the first unsupervised approach from the classifications of observations prior to each failure in the test sets.

	S1		*S2*	
	27.10.2018	01.01.2019	27.10.2018	01.01.2019
Total number of hours classified	12	11	12	11
Anomalies detected	1	0	2	0

Table 5.2: The table presents the results of the hypothesis testing for the first unsupervised approach.

Actual number of failures observed	6
Failures with a statistically significant number of anomalies detected	0

Table 5.1 and 5.2 presents the obtained results based on the definitions from Section 4.6. Table 5.2 shows that non of the failures had statistically significant observation prior to the failures in any of the test sets. However as one can observe in Figure 5.3, the predictions from *S2* 27.10.2018 shows two consecutive anomalies some hours before the fault, given that the NU parameter is set to 0.05, one can expect 5% of healthy observations to be detected as anomalies, the probability of observing two or more anomalies in 12 healthy trials is 11.8%. However, as we defined our significance level to be 1%, 11.8% is not within our rejection area.

Table 5.1 shows that a total of three out of 46 hours prior to faults were classified as anomalies to the normal condition, while 43 hours were considered healthy. In order to determine if these results are significant, we need to calculate the probability of seeing three anomalies in a healthy dataset. As the NU parameter is set to 0.05, the expected number of anomalies observed in a healthy dataset is given by Equation 5.1, with $X \sim B(2.3)$.

$$E(X) = 0.05 * 46 = 2.3 \tag{5.1}$$

$$P(X \geq 3) = 1 - P(X < 3) \tag{5.2}$$

$$P(X \geq 3) = 1 - [P(X = 2) + P(X = 1) + P(X = 0)] \tag{5.3}$$

$$P(X \geq 3) = 0.406 \tag{5.4}$$

As there is approximately 40% chance of obtaining three or more anomalies from a random sample of "healthy" data (data from the training set), the results cannot be considered statistically significant, thus we cannot reject the null hypothesis.

Second Unsupervised Implementation

In the second unsupervised approach, we trained four separate prediction models. Each of the prediction models stems from training data at a specific date and substations.

Figure 5.5 shows the distributions of the scoring functions for the four prediction models. When inspecting the test set curves one can notice two interesting observations. The upper right plot includes the scoring function distribution for the test set of S2 on 27.10.2018. The left-most peak is due to the actual faults that day. However, there is also a very distinct peak closer, from the vertical decision boundary line, belonging to the anomalies detected between 04:00 and 06:00. The other interesting observation is seen in the bottom left plot, showing the distributions of test and training set for *S1* 01.01.2019. We observe that the test set is noteworthy right-shifted compared to the training set.

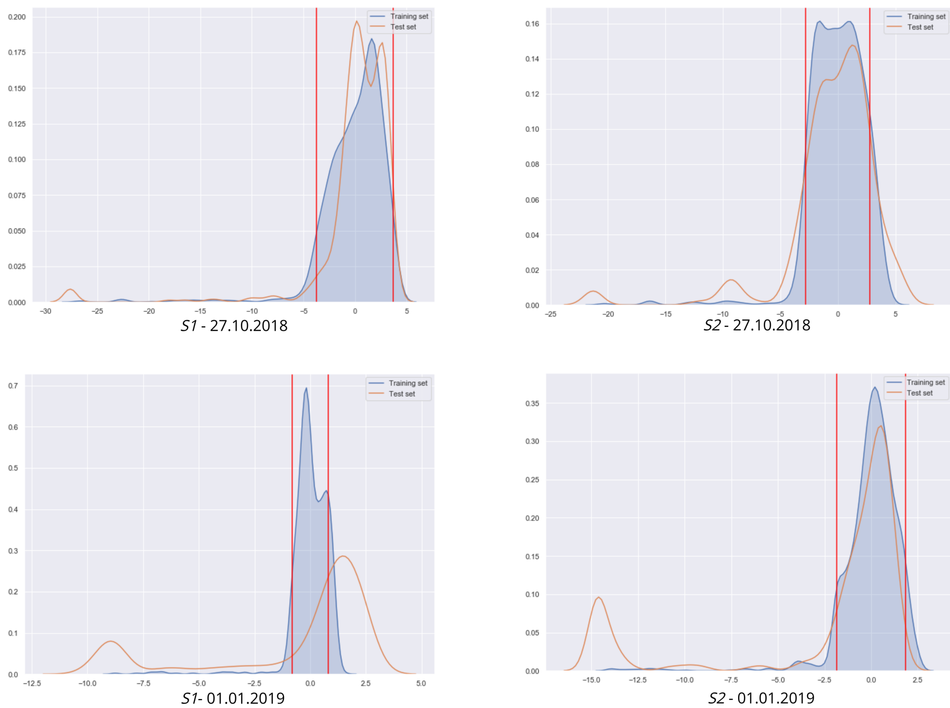

Figure 5.5: Distribution comparison plot with data for all four training and test sets.

Table 5.3: The table presents the results from the second unsupervised approach.

	S1		*S2*	
	27.10.2018	01.01.2019	27.10.2018	01.01.2019
Total number of hours classified	154	610	154	369
Anomalies detected	6	52	4	29
$P(X \geq x)$ (P-value)	0.7797	0.0003	0.9482	0.0139

Table 5.4: Hypothesis Test Results for the second unsupervised prediction model, using minute values.

Actual number of failures observed	8
Failures with a statistically significant number of anomalies detected	2

Table 5.3 and 5.4 shows the results related to the second unsupervised approach. Table 5.4 shows that there are eight failures in the training sets in the second unsupervised model, while the first unsupervised model only included six. This is because two of the faults occurred within the same hour.

Table 5.3 show that a total of 91 out of 1287 observed minutes prior to outages, are marked as anomalies to the normal condition. Hence, the model predicts $91/1287 \approx 7\%$ anomalies. Given the nu-value of 0.05, we expect 5% of healthy observations to be predicted as anomalies. The probability of obtaining 7% anomalies out of randomness is computed by calculating the cumulative probability for P(X $\geq$ 91). When calculating this, according to the Poisson distribution formula, we get a *p-value* of 4.6×10^{-4}, which indicates that predicting such a large number of anomalies out of pure randomness is 0.046%. According to these results, there is evidence to suggest that the percentage of observations prior to an outage marked as anomalies is greater than 5%.

Table 5.4 indicates that two of the faults observed in the test set had statistically significant observations before the failure occurred. Both of the statistically significant observations belong to faults which occurred January 1, 2019. Both *S1* and *S2* had a statistically significant number of anomalies detected during the hours prior to this fault. Table 5.5 show all the faults p-value scored on the hypothesis testing. The scores are calculated using Poisson distributions. Two of the *p-values* in Table 5.5 are smaller than the set significance level of 0.01.

Table 5.5: Obtained *p-value* in hypothesis test results for the second unsupervised prediction model, using minute values. With H_0 being that 5% of the observations prior to a failure is classified as anomalies, and the *p-value* is the probability of finding the observed, or more extreme, results when the null hypothesis (H_0) is true.

Fault	*S1*	*S2*
27.10.2018 (12:34-12:40)	0.7797	0.9482
01.01.2019 (10:33-10:35)	0.00001	0.3587
01.01.2019 (11:01-12:07)	0.7135	0.7135
01.01.2019 (14:50-14:51)	0.5609	0.0008

5.2 Interpretation of Results

In this section, we review, interpret, and discuss the results presented in the previous section. We evaluate the results in terms of validity and substance before we review and discuss the results in accordance with the research questions. We present the interpretation of the results in the same order of which they were obtained, starting with the supervised approach, followed by the two unsupervised approaches.

The results obtained by the supervised approach gives a perfect confusion matrix, presented in Figure 5.1. The perfect confusion matrix is a result of the model overfitting to the observations from the training set. The model was tested on the same faults of which it was trained. The results from this approach do, as previously mentioned, not replicate how a model would behave to previously unseen data.

Hence, the supervised results do not provide any real-world value — however, the method of labeling the data with a sliding window technique. In combination with a random forest classification method, proves its ability to recognize patterns. Such a model with extensive amounts of data could provide valuable results for predicting grid faults, and might even be able to classify the type of fault observed if given a large enough training set. The approach followed in the supervised model could serve as an inspiration for future work.

On the other hand, the results obtained by the unsupervised models presented in this thesis are more exciting. We start with an interpretation of the results obtained from the first unsupervised implementation. Already in the distribution plots of the score function between the training set and the test set, in Figure 5.2 there are interesting observations to be made. The first is that the test set from *S2* looks more skewed to the left than that of *S1*. Which might indicate that the substations have individual attributes, that may play a part in how the test set compares to the training set.

Secondly, the wideness of the score distribution of the test sets indicates that their score function contains a more significant variance than that of the scores from the training sets. The significant variance might imply that the condition in the substations is more fluctuating during the faulty days. The predictions for *S1* and *S2* presented in Figure 5.3 from 27.10.2018 show that the model observes anomalies for both *S1* and *S2* prior to the fault. At first glance, the observation from *S1* in Figure 5.3 looks promising, as it observes an anomaly during the hour prior to the fault. Such observations are what we want a predictive maintenance model to consistently provide. Nevertheless, this observation could be a result of randomness, as 5% of the healthy training data are classified as anomalies.

Moreover, one may discuss the healthiness of the training data and argue that a predicted anomaly means that the observation is outside 95% of the training sets hypersphere. Thus, the training data may not be entirely representative of a healthy dataset; for example, there might be anomalies in the training set, which has not yet resulted in a fault. This would imply that the anomalies detected in the training set are real. Observing *S1* in Figure 5.2 the curve of the test set looks rather normally distributed. However, the small bump to the left of the red decision boundary demonstrates that the anomaly is rather significant compared to the rest of the test set. Which might indicate that anomaly detected the hour prior to the fault did not occur by chance.

Table 5.1 shows that out of the 46 hours observed before the faults 27.10.2018 and 01.01.2019, only three of these were classified as anomalies, not a very promising result. However, the fact that all three observations stem from the same date could be an indication that this fault is possible to predict with the data at hand. While the fault observed 01.01.2019 is not. Also, the data resolution might be decisive of what types of errors are predictable.

As our results show, the test sets with hour resolution provides better results for the faults that took place on October 27. While the minute resolution used in the second approach provides better predictions on the faults from January 1. Knowing the root cause of the faults at hand would be helpful in order to better understand the relationship between sample resolution and classification of faults. Further analyses of this topic are required.

For the second unsupervised implementation, we witness some interesting observations in Figure 5.5. Figure 5.5 shows the scoring functions for the four prediction models. The prediction model from *S1* on 01.01.2019 is right-shifted compared to the training set. The other three test sets show quite similar distributions as their training set. However, some anomalies in the form of small bumps to the left of the training set distribution are possible to observe in the Figure 5.5.

The fact that the distribution of the test set is right-shifted compared to the training set, in *S1* from 01.01.2019 in Figure 5.5, may be a result of the model training on unhealthy data. It is hard to determine how long prior to an outage anomalies starts to occur. According to [RBC+09], mentioned in the Related Work chapter, anomalies may be observed intermittently for at least six days prior to an outage. Thus all training sets used in this approach might already contain a large number of anomalies, which contributes to a wrongly trained model.

Another indication that the training sets might contain unhealthy data is the small bumps which might be observed left of the training sets main distributions in Figure 5.5. These bumps indicate anomalies in the training data. The bumps are apparent in all four distributions. The idea of training and testing the model on data from the same substation is an excellent approach for incorporating individual differences between substations into the model. However, it is unfortunate that we obtained what possibly already is "unhealthy" data from the same day as a failure occurred to the dataset.

Table 5.4, which belongs to the second unsupervised approach, indicates that two out of eight of the faults had statistically significant observations prior to the fault in the test sets. Both of the faults indicated as statistically significant belong to the same day (01.01.2019). One of the statistically significant observations occurs before the first, of many, fault this day. The fact that there were repeating failures this day may indicate that the DSO did not manage to get the situation under control. So the fault might be a result of a harmful component, which required staff to inspect and replace physically.

If the fault from 01.01 was a result of a predictable fault. Then it is difficult to argue that the training set does not contain any faults. If the training set contains many anomalies, the results do not replicate "real-world" results, which might have yielded better results with a sufficient dataset. The two failures with a statistically significant number of anomalies detected presented in Table 5.4 rejects the null hypothesis, by a large margin. These results are extraordinary but might be too good to be true, as this could be a result of training on an unhealthy dataset.

Table 5.3, shows that out of the 1287 minutes inspected prior to failures, the model detects a total of 91 anomalies. These results are statistically significant. This may be an indication of large variances in the test sets compared to the training sets. However, a too small training set compared to the training set could result in many anomalies. We trained on data from approximately 16 hours a day and tested on data from eight hours a day. A better approach might be to have a train-test ratio of 80-20. It is also worth noting that the results would not be as remarkable if it were not for the anomalies observed prior to the two statistically significant failures.

The results from the two unsupervised approaches are exciting and show somewhat different results. The first unsupervised model predicts better on the fault from 27.10, while the second approach showed significant results on the fault from 01.01. Perhaps a result of the different resolutions used in the two approaches, maybe in combination with the type of fault that occurred, unfortunately data on what caused the failure is not available to us. Therefore, one cannot draw definite conclusions from these results, due to the lack of sufficient amounts of data.

It should be of interest to Grid operators in Norway to invest more resources into research related to predictive maintenance in grid operation. Providing researchers with necessary data and domain knowledge to come up with more precise results into how best to detect and classify failures in the grid. Our results imply that there are potential using sensor data in combination with machine learning for predictive maintenance in operations of the Norwegian power grid. However, due to the lack of data, a finite conclusion cannot be drawn from these results.

5.2.1 Assumptions

After collecting the data, we made several assumptions on the behavior of the system studied. In order to evaluate the validity of our results, it is necessary to state the assumptions made when conducting the experiments and consider how these compare to the real world. While some of these assumptions are reasonable, others might differ from reality.

During the supervised approach, we combined observations from *S1, S2*, and *S3* to build a single prediction model, without including any individual attributes of the substations in the feature-set. Relevant individual attributes could be the number of customers connected to a substation, age of the substation, or suppliers of components. No such information was included in the datasets. Hence, we regard these substations as equal and implicitly assume no individual differences between them.

In the first unsupervised approach, we trained a prediction model on data from *S3*, to make predictions on *S1* and *S2*. Hence, the same assumption as in the supervised approach is made, that there are no individual differences between *S1*, *S2*, and *S3*. As we never received any information on individual attributes of substations, it is difficult to compare this assumption to the real world.

The second unsupervised approach takes individual differences between substations into account. By building separate prediction models for *S1* and *S2*, each trained only on observations from themselves. In this approach, we trained the prediction model on observations collected from a day containing faults. An observation from a substation is included in the training set as long as no faults occur less than three hours before or after that observation is made. Hence we assume that faults do not cause anomalous measurements for more than three hours in advance. This assumption was necessary to make due to the limited availability of observations from prolonged periods of normal condition.

As described in the section on Data Collection, we were not able to collect information on the root causes of the faults in our dataset from *S1* and *S2*. As a result, we cannot be sure that the faults we are trying to predict are possible to predict. Hence, for all approaches, we also assumed that the faults which occurred are possible to predict.

Some of the assumptions we made are difficult to compare to real-world behaviour, as more domain knowledge is needed to make these comparisons. Although such domain knowledge could have strengthened our results, we do not consider the absence of this knowledge to be decisive in the validation of the results.

5.2.2 Substance

We differentiate between the substance and validity of the results obtained through the experiments performed and the substance of the research as a whole.

The results obtained are promising. However, due to the lack of sufficient amounts of data and metadata, we made assumptions about the data, and cannot draw any valid conclusions on which ML algorithm that performs better or how one should implement such an alarm system with live data from sensors. Nor, to what extent prediction is possible. Hence, the substance of the results obtained is limited.

However, the fact that predictive maintenance in the Norwegian power grid has not been tested before. In combination with results showing significant observations on such a limited dataset should convince operators to be more willing to share data and domain knowledge with researchers.

The main contribution of our work is that we provide a fundamental for utilizing historical sensor data for constructing ML algorithms for maintenance purposes in the Norwegian power grid. We have paved the way for others wanting to conduct similar studies, so they are better prepared to face the challenges we faced, learn from the background research conducted and may utilize our approaches as a framework.

5.3 Review of Research Questions

In this section, we look back at the research questions stated in the introduction. We have researched and conducted experiments to provide answers to the questions listed below.

1. Why should machine learning and predictive maintenance be of interest to Norwegian grid companies?

The answer to the first research question is that the massive installation of new data sources provides operators with more data. The increase in available data in combination with cheap computational power provides an opportunity for grid operators to look into how best to utilize the data they now have at hand. The fact that Norwegian operators pay hundreds of millions NOK per year in CENS, for unplanned outages. Combined with results from our and other related research, which suggests that prediction is possible on some types of faults, should be enough to draw DSOs attention towards this field of study.

Successful implementation of predictive maintenance would improve asset management, decrease the percentage of unplanned outages. Planned outages may be performed during beneficial hours, such as during nighttime, rather than having unplanned outages during "rush-hours" when more people are affected. If one could predict and thus prevent an outage – this would ensure increased availability, more reliable operation, and improved safety.

2. Which external data sources can be combined with grid data to increase insight on the distribution grid?

Regarding the second research question, we found that the literature mostly suggests using exogenous variables, such as weather data, especially wind and rain. Hence, we implemented our solution accordingly. Additionally, metadata such as the root cause for failure could also be helpful. Further, substation specific information, such as the number of connected customers, age of various components, and mean time between failure of components would also increase insight. However, due to the limited data at hand, we did not get the chance to test whether such metadata would improve the accuracy of such an algorithm. Furthermore, due to the general uncertainty of our results, we did not find it useful enough to test the effect of implementing ML algorithms with versus without weather data.

3. What is the potential of using sensor data and machine learning techniques for predictive maintenance in operation of the Norwegian distribution grid?

As for the third research question, we were not able to draw an exact conclusion due to the lack of data. However, after the research conducted in this thesis, we believe that components do give away early signs of deterioration, if operators monitor the right features, prediction should be possible. However, due to the limited data and the assumptions we were forced to make, we cannot provide good enough evidence to back up this hypothesis. On the other hand, the results does not conclude the opposite, hence there is still reason to believe that based on data provided from sensors already installed in the power grid, prediction is possible.

5.4 Challenges and Limitations

In this section, we address the challenges and limitations faced during the work on this thesis. The primary challenge faced during this thesis has been the lack of sufficient amounts of data. Further, the lack of proper domain knowledge has lead to challenges during the process. Additionally, an assumption we have made is that data from our DSO are similar to that of other DSOs. Hence, we assume that grid data collected from all providers are similar, which is a limitation to our study.

5.4.1 Process of Data Collection

As described in chapter 4, the process of collecting data from DSOs proved to be a more significant challenge than initially expected. Firstly, there was the challenge that the operators did not have sufficient amounts of faults in the substations containing sensors. Additionally, there were challenges when trying to export the collected data.

During the last 12 months of operation, only three out of the approximately 30 sensor-instrumented substations experienced faults. Ideally, we would have been able to collect data containing hundreds of faulty observations, as well as prolonged periods with normal condition data. When we became aware of the rareness of faults in their dataset, we requested data from more extended periods with healthy observations. Unfortunately, technical difficulties prevented the DSO from being able to export data of such quantities. However, the DSO was able to share data from the year 2018 from *S3*, as they had previously exported this data for other purposes. The process of exporting more extended periods of data from the DSOs SCADA-systems, where the sensor measurements are stored was slow. According to the DSO, this process took up to weeks and included manual labor for exporting the chosen data to excel files. We are unaware of why this task is such a demanding one.

We underestimated the process of extracting data for operators. Also, it was our impression was that operators held data from more faults than what they actually did. The fact that there were difficulties extracting the data, indicates that implementing PdM might not be the next step in making the grid smarter for DSOs. Rather, DSOs should first focus on support for remote controlling and self-healing functionality which operators currently are researching.

When remote controlling and self-healing elements are in place, the operators will be running algorithms based on real-time sensor data. Next, implementing a well-functioning PdM program should be on their agenda. However, before this, an optimal ML algorithm should be in place. Thus there should be research into how data is stored. The most critical criteria for data storage is the security element. However, ease of data retrieving should be of top priority in order to simplify data collection for further work on topics such as predictive maintenance.

It is apparent that DSOs does not collect data for PdM. However, DSOs should have an idea of the purpose of their data collection. Awareness of why they collect data would give a better idea of what resolution of data to collect and where to best store it.

5.4.2 Quality and Quantity of Collected Data

Scarcity of failure event data conditioned the effectiveness of the employed machine learning algorithms. Preferably we should have had somewhere between 200 and 2000 interruptions rather than 8, for implementing a ML algorithm which could provide value to the DSO. However, this was limited by the number of substations which are instrumented. Thus the number of faults the DSO has measurements from, as well as the difficulties for the DSO to extract the data which they had measured.

We obtained limited amounts of observations. Hence, it was difficult to know whether the observations we used for training the model were truly healthy. In the unsupervised approaches, we assumed parts of the data from *S1* and *S2* were healthy, although these observations were registered in periods close to grid faults.

Another challenge faced during the process was that the various substations, track different features. For instance, one of the substations tracks oil-temperature, while the other two does not. The process would be significantly easier if all substations had measurements of similar features. The fact that various substations measure different things resulted in that we had to drop some features, such as a power measurement which seemed relevant; however, as it was not present in all substations, we had to drop this feature.

Furthermore, substation information could be relevant, such as the number of connected substations or customers. As well as the age of the substations, the time elapsed since last maintenance, and length of incoming and outgoing lines. We expected that such information added to the features could potentially improve the model.

5.5 Evaluation of Methodology

This section discusses the methodology used throughout the thesis. First, we discuss the methodology around the conversations with the DSO. Then we discuss the approaches taken in the technical implementation.

5.5.1 Conversation with the Distribution System Operators

The conversations with the DSOs were performed in a manner of semi-structured interviews. We underestimated the difficulties of obtaining data from DSOs, and thought that sending emails to two of the larger DSOs in the CINELDI project should be sufficient. In the emails, we explained our plan and thoughts to why we believed that researching PdM would be of interest to them as well as being open for suggestions for other use-cases of ML in the power grid.

One of the DSOs was quick with sharing some data, and we sat up a live meeting for discussing the meaning of the shared data. During this meeting, the DSO expressed their concern toward us not being able to predict "unpredictable" events, such as birds damaging the lines and water entering the pipes. However, the DSO stated that they would try to fetch more data for our purpose, however, this could take some time, as it required manual labor. Nevertheless, after some months, we received an email stating that they were not able to download any more data, as there were challenges regarding the access to their SCADA system.

We should have done a more thorough job preparing conversations with multiple DSOs. An idea would be sending a questionnaire to multiple DSOs for surveying which areas of research the DSOs believe that best utilize the data collected from the sensors. Furthermore, more physical meetings to discuss the problem area would be beneficial. In the following parts of the methodology evaluation, we discuss the technical approach taken in this thesis.

5.5.2 One Complex Model vs Many Separate Models

When applying ML to solve problems, one usually prefers to train one general model, rather than to tailor many models to specific tasks. A notable contrast between the two unsupervised approaches in our experiment was that in the second implementation, we build separate prediction models for each substation. Thus, rather than building one model for all substations, we build one per substation. With the characteristics of the data collected in this thesis, the latter seems like the most reasonable approach. However, when the accessibility of measurement data and individual attributes of substations improves, the choice between these approaches may change. This choice depends mainly on the amount and type of data available. While it is difficult to foresee what type of data that will become available in the future, suitable approaches for different cases may still be discussed.

First, we consider the approach of training one model on data from multiple substations. Accuracy of a prediction model based on unsupervised learning is highly dependent on being built by using a training set that is representative of the measurements observed during normal condition. For a model trained on observations from substation A to be able to make predictions on substation B, the individual differences of the substations need to be negligible, or the individual attributes need to be included in the feature-set. It is unlikely that the individual differences between two randomly selected substations are small enough to be considered negligible. Hence, the inclusion of individual attributes of the substations in the feature-set is necessary for building a common prediction model.

Now we consider the second approach, which the second unsupervised implementation in our experiment uses. In this case, each of the substations trains only on data from itself. Because all measurements now stem from the same substation, no individual attributes needs to be included in the feature-set. An advantage of this approach is that the prediction model will be less complex than a common model, and it will be easier to understand the rules that it uses. Naturally, a prerequisite for using this approach is that data from the specific substation already exists. This means that for new substations, an extended period of collecting measurement data is needed before a model can be fitted to the substation.

Although building separate models seems to be the most reasonable approach in our experiment, a common model might have more advantages in other cases. The fact that faults in substations occur as rarely as they do suggests that building a common model could be more feasible than separate models. Also, the implementation of a common model in a new substation could be rapidly set up as the model already exists. The same process would take more time with separate models. However, the approach of a common model requires individual attributes of each of the substations to be known. Before such a model is built, an analysis should be conducted to research what these attributes need to describe about a substation. Without access to these attributes, the potential of using a common prediction model is reduced.

5.5.3 Considerations of Technical Implementation

This section discusses some of the choices made during the technical implementation of the experiment and how these affect our results, as well as alternative choices. Also, we discuss resolution and trends in the measurements.

The first and second conducted experiment used training data with hourly aggregated measurements to build a prediction model. While in the third experiment, the training data contained measurements aggregated over one minute. The decision on what time resolution is most relevant for the prediction of component failure depends on the component and type of failure. Time granularity is critical when trying to detect intermittent failures leading to permanent faults. A component might show patterns, indicating imminent failure; however, when aggregated at a resolution of one minute or one hour, these patterns might no longer be detectable. At the same time, other components might require the measurements to be aggregated at a resolution of one hour, or even more, for showing patterns indicating failure to be detected by a prediction model. Thus, an optimized prediction model needs to be able to detect patterns appearing at different levels of resolution.

Patterns indicating upcoming failures may appear as particular combinations of the different measured values in one observation, or they may appear when inspecting consecutive measurements of specific values. For detecting the latter, looking at multiple consecutive observations at the same time is necessary. In the models built in our experiments, consecutive observations were not inspected together. As an example, the effect from a heavy rainfall might have a significant delay from the time the rainfall happens, until the time when underground cables may be affected.

Hence, these models will not be able to detect patterns appearing over multiple observations. For the first and second experiment, only patterns appearing inside one hour of measurements may be picked up. For the third experiment, the model may only recognize patterns appearing inside one minute of measurements. Although limited, the observations provided to our models include aggregated values of minimum, maximum, and average over the respective time resolution. Thus, some patterns appearing inside the given time intervals may still be picked up. Nevertheless, providing more aggregated statistics from each time interval, like variance, is expected to increase the chances of the model learning other patterns than the ones learned in our experiments.

ML is a field with numerous use cases related to the power grid. Although ML has seen rapid development during the last decade, the need for domain knowledge when trying to solve specific problems will always be crucial. Previously, we mentioned the challenges of choosing an adequate resolution of aggregation, as well as understanding what data is needed for detecting different patterns which may indicate faults. Technical knowledge from the power grid domain would provide beneficial insight into the process solving such challenges. In the same way, as domain knowledge is valuable for building better prediction models, more domain knowledge is needed to interpret the performance of the models. After our experiment, understanding and deciding exactly how to interpret the results suitably was time-consuming. Different metrics for defining the accuracy could have been used. For example, the predictions made by the model could have been weighed heavier for observations closer to the time of the actual fault.

Chapter 6
Conclusions and Further Work

6.1 Conclusion

Successful implementation of a predictive maintenance program should be of interest to Norwegian grid operators, as this could substantially reduce the cost of operation for DSOs. Predicting grid failures and having a crew replace faulty components before they result in outages will improve grid reliability and security of electricity supply. Henceforth, the implementation of such a solution should be a priority for grid operators.

We have presented two machine learning models using unsupervised learning techniques. The purpose of these models is to detect anomalous measurements in power grid data. Additionally, we created a framework for building machine learning models through a supervised approach. Such models may be used for predicting upcoming failures in the power grid.

We found that the unsupervised approach using a one-class SVM for anomaly detection, with minute resolution, provided the most promising results. This algorithm marked a statistically significant number of anomalies prior to two of the eight failures observed. The relatively small number of failures in the power grid in general, might indicate that an algorithm trained to distinguish normal condition data from faulty condition data could be useful when operating the power grid. Such an anomaly detecting algorithm could be implemented and used in an alarm system, to alert personnel of situations where substations should be manually inspected.

Due to limitations in the data obtained, we were forced to make several assumptions when creating the two algorithms and the framework. These limitations concerned missing information about individual differences between substations and the root causes of the failures. Limited knowledge on the behavior of faulty components prior to a failure required further assumptions during implementation. Research remains incomplete as we were not able to demonstrate that practical applications of a predictive maintenance solution are precise enough to be trusted.

We propose that data which may be used for predictive purposes are stored in easily accessible databases. Such data include sensor measurements and information about the cause of failures. Solutions for improving data accessibility will make it easier for operators to participate in future similar projects. Also, a joint effort and close collaboration between analysts and domain experts, would be beneficial in the process of selecting parameters which may indicate deterioration of components.

6.2 Future Work

Future work should implement similar models with larger datasets containing significantly more failures than ours. In order to obtain such datasets, future work could collect data from multiple DSOs. Further, failures should be labeled with the root cause, if such information is available. Implementation of a trained model, able to operate with real-time weather and grid data, would be a significant contribution towards providing tools for intelligent decision support to the grid operators.

Statnetts measure of instantaneous balance in the power system should be further explored. Information on sudden changes in electricity production or consumption may provide valuable insight for building similar prediction models. Historical data on instantaneous balance was not available at our hand during the creation of the models in our experiments.

Technically, future work should investigate how trends in sensor measurements can be analyzed to improve the model. For instance, measurement statistics from the last minute, hour, week and month could be included in the same feature-set. To provide a broader knowledge of how failing components may be detected, many alternative ways of analyzing trends should be explored. Also, an investigation of the optimal resolution of grid measurements for predictive maintenance is needed.

Furthermore, future work should not underestimate the challenges related to obtaining sufficient amounts of data, and should realize the value of having close collaboration with DSOs.

References

[AK] Jorgen Aarstad Anders Kringstad, Vegard Holmefjord. Fleksibilitet i det nordiske kraftmarkedet. https://www.statnett.no/globalassets/for-aktorer-i-kraftsystemet/planer-og-analyser/2018-Fleksibilitet-i-det-nordiske-kraftmarkedet-2018-2040. [Online; accessed 19-March-2019].

[Anu18] Anup Bhande. What is underfitting and overfitting in machine learning and how to deal with it. https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76, 2018. [Online; accessed 04-April-2019].

[ATHU18] Christian Andresen, Bendik Nybakk Torsaeter, Hallvar Haugdal, and Kjetil Uhlen. Fault detection and prediction in smart grids. In *2018 IEEE 9th International Workshop on Applied Measurements for Power Systems - AMPS*, pages 1–6, 09 2018.

[Bar] Bartz/Stockmar. Staying big or getting smaller. https://commons.wikimedia.org/w/index.php?curid=69505750. [Online; accessed 28-May-2019].

[BBM15] Simona Bigerna, Carlo Andrea Bollino, and Silvia Micheli. Overview of socio-economic issues for smart grids development. In *Proceedings of the 4th International Conference on Smart Cities and Green ICT Systems - Volume 1: SMARTGREENS,*, pages 271–276. INSTICC, SciTePress, 2015.

[BLB+13] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, and Gaël Varoquaux. API design for machine learning software: experiences from the scikit-learn project. In *ECML PKDD Workshop: Languages for Data Mining and Machine Learning*, pages 108–122, 2013.

[CCV08] Sivia Cateni, Valentina Colla, and Marco Vannucci. *Outlier Detection Methods for Industrial Applications.* 2008.

[cin] Centre for intelligent electricity distribution. https://www.sintef.no/projectweb/cineldi/. [Online; accessed 03-October-2018].

[Con16] J Conway. The industrial internet of things: an evolution to a smart manufacturing enterprise. *Schneider Electric*, 2016.

[DO18] Demilade Dinakin and Peter Oluseyi. Optimal under-frequency load curtailment via continuous load control in a single area power system using fuzzy logic, pid-fuzzy and neuro-fuzzy (anfis) controllers. 4:208–223, 12 2018.

[DY14] Li Deng and Dong Yu. Deep learning: Methods and applications. Technical Report MSR-TR-2014-21, May 2014.

[Ene19] Energy Facts Norway. Security of Electricity Supply. https://energifaktanorge.no/en/norsk-energiforsyning/forsyningssikkerhet/, 2019. [Online; accessed 12-March-2019].

[ES18] Nils Stieglitz Erik Solheim, Patricia Espinosa. Global trends in renewable energy investment 2018. 2018.

[EYSKBL17] IEEE Gerald J. FitzPatrick Member IEEE Eugene Y. Song, Member and IEEE Kang B. Lee, Life Fellow. Smart sensors and standard-based interoperability in smart grids. *IEEE Sensors Journal*, 17(23), Juni 2017.

[For] Forbes. A Short History of Machine Learning. https://www.forbes.com/sites/bernardmarr/2016/02/19/a-short-history-of-machine-learning-every-manager-should-read/. [Online; accessed 22-May-2019].

[GKWK15] S. Gupta, R. Kambli, S. Wagh, and F. Kazi. Support-vector-machine-based proactive cascade prediction in smart grid using probabilistic framework. *IEEE Transactions on Industrial Electronics*, 62(4):2478–2486, April 2015.

[GSK+11] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati, and G. P. Hancke. Smart grid technologies: Communication technologies and standards. *IEEE Transactions on Industrial Informatics*, 7(4):529–539, Nov 2011.

[HA17] Eric Hsieh and Robert Anderson. Grid flexibility: The quiet revolution. *The Electricity Journal*, 30(2):1 – 8, 2017.

[Har12] Peter Harrington. *Machine Learning in Action.* Manning Publications Co., Greenwich, CT, USA, 2012.

[Hat01] Nikolaos and Hatziargyriou. *Machine Learning Applications to Power Systems*, pages 308–317. Springer Berlin Heidelberg, 2001.

[IEC19] International Electrotechnical Commission 617-01-06. http://www.electropedia.org/iev/iev.nsf/display?openform&ievref=617-01-06, 2019. [Online; accessed 06-June-2019].

[Inn19] Innovasjon Norge. Innovasjon Norge kundehistorie - Tibber AS. https://www.innovasjonnorge.no/no/tjenester/kundehistorier/kundehistorie/tibber-as/, 2019. [Online; accessed 19-February-2019].

[Kir19] Kirill Eremenko, Hadelin de Ponteves, SuperDataScience Team. Machine Learning A-Z™: Hands-On Python & R In Data Science. https://www.udemy.com/machinelearning/, 2019. [Online; accessed 2-February-2019].

[KM09] Shehroz Khan and Michael Madden. A survey of recent trends in one class classification. *Artificial Intelligence and Cognitive Science, Lecture Notes in Computer Science, vol. 6206*, 6206:188–197, 08 2009.

[KS15] Poul Heegaard Kjell Sand. Next generation control centres – state of art and future scenarios - version 2.0. October 2015.

[Laz] Levelized cost of energy 2017. https://www.lazard.com/perspective/levelized-cost-of-energy-2017/. [Online; accessed 27-April-2019].

[LO] Sajan Bhantana Lovinda Oedegaarden. Status og prognoser for kraftsystemet 2018. http://publikasjoner.nve.no/rapport/2018/rapport2018_103.pdf. [Online; accessed 8-March-2019].

[Lov] Forskrift om produksjon, omforming, overføring, omsetning, fordeling og bruk av energi m.m.(energilovforskriften). *LovData*.

[LSKM04] Pavel Laskov, Christin Schäfer, Igor Kotenko, and Klaus-Robert Müller. Intrusion detection in unlabeled data with quarter-sphere support vector machines. pages 228–236, 12 2004.

[meta] How to use the frost api. https://frost.met.no/howto.html. [Online; accessed 04-May-2019].

[metb] How to Use the Frost API - Python Examples. https://frost.met.no/python_example.html. [Online; accessed 04-May-2019].

[MHH18] Eivind Skjærven Torunn Høstad Sliwinski Silje Cathrine Syvertsen Lars Varden Mona Helen Heien, Tore Langset. Forslag til endring i forskrift om kontroll av nettvirksomhet. *NVE*, 2018.

[Mob04] R. Keith. Mobley. *Maintenance Fundamentals*. Elsevier, Burlington, 2004.

[NAS] NASA. The causes of climate change. https://climate.nasa.gov/causes/. [Online; accessed 27-March-2019].

[Nau15] Dewang Nautiyal. Top 5 best programming languages for artificial intelligence field. https://www.geeksforgeeks.org/top-5-best-programming-languages-for-artificial-intelligence-field/, 2015. [Online; accessed 07-May-2019].

[Nor19] Energy Facts Norway. Electricity Production. https://energifaktanorge.no/en/norsk-energiforsyning/kraftproduksjon/, 2019. [Online; accessed 12-March-2019].

[nve] https://www.nve.no/energy-market-and-regulation/network-regulation/. [Online; accessed 02-February-2019].

[oE10] US Department of Energy. Operations & maintenance best practices guide: Release 3.0. *FEDERAL ENERGY MANAGEMENT PROGRAM*, 2010.

[oxf] Definition of overfitting in english by oxford dictionaries. https://en.oxforddictionaries.com/definition/overfitting. [Online; accessed 09-February-2019].

[p-v] P values. https://www.statsdirect.com/help/basics/p_values.htm. [Online; accessed 23-May-2019].

[PD11] Peter Palensky and Dietmar Dietrich. Demand side management: Demand response, intelligent energy systems, and smart loads. *Industrial Informatics, IEEE Transactions on*, 7:381 – 388, 09 2011.

[PMG04] Kjetil Ingeberg Eva Næss Karlsen Jon Sagen Karina Veum Paul Martin Gystad, Carl-Petter Haugland. Prinsipper for regulering av nettvirksomhetens inntekter. *NVE*, 2004.

[RBC+09] B. D. Russell, C. L. Benner, R. M. Cheney, C. F. Wallis, T. L. Anthony, and W. E. Muston. Reliability improvement of distribution feeders through real-time, intelligent monitoring. In *2009 IEEE Power Energy Society General Meeting*, pages 1–8, July 2009.

[RLS14] Michele Ruta, Giuseppe Loseto, and Simone. Try to predict grid faults: A dynamic, semantic-based and multi- dimensional approach. 2014.

[RWA+12] C. Rudin, D. Waltz, R. N. Anderson, A. Boulanger, A. Salleb-Aouissi, M. Chow, H. Dutta, P. N. Gross, B. Huang, S. Ierome, D. F. Isaac, A. Kressner, R. J. Passonneau, A. Radeva, and L. Wu. Machine learning for the new york city power grid. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(2):328–345, Feb 2012.

[sin] Security of electricity supply. https://www.sintef.no/en/security-electricity-supply/. [Online; accessed 29-May-2019].

[SSB18] https://www.ssb.no/energi-og-industri/statistikker/elektrisitet/aar, 2018. [Online; accessed 07-May-2019].

[Sta17] Statnett. System operations and market development plan 2017-2021. September 2017. Accessed: 2019-02-20.

[Suh15] E. Suhir. Aging-related failure rate obtained from bathtub curve data. In *2015 IEEE Aerospace Conference*, pages 1–8, March 2015.

[Tje16] Jørgen Tjersland. Estimering av kile-kostnader ved seksjonering og feilretting i høyspennings distribusjonsnett. 2016.

[TL13] John G. Cock Siri H.Steinnes Tore Langset, Thor Martin Neurauter. Endringer i forskrift om kontroll av nettvirksomheten. *NVE*, December 2013.

[Tow] Model evaluation techniques for classification models. https://towardsdatascience.com/model-evaluation-techniques-for-classification-models-eac30092c38b. [Online; accessed 08-May-2018].

[VCG10] Gerhard P. Hancke Vehbi C. Gungor, Bin Lu. Opportunities and challenges of wireless sensor networks in smart grid. *IEEE Transactions on Industrial Electronics*, February 2010.

[voe18] Norges vassdrags-og energidirektorat. Avbrotsstatistikk 2017. June 2018.

[VVM+16] J. L. Viegas, S. M. Vieira, R. Melício, H. A. Matos, and J. M. C. Sousa. Prediction of events in the smart grid: Interruptions in distribution transformers. In *2016 IEEE International Power Electronics and Motion Control Conference (PEMC)*, pages 436–441, Sep. 2016.

Appendix A
Python Code

The appendix includes some of the code used for implementing the machine learning algorithms. The code was modified during the process to accommodate for the various algorithms we made. However, presented below is some of the code used throughout this project.

A.1 Import and Reform Dataframes From *S1*, *S2* and *S3*

```
import numpy as np
#Importing S3 dataset
df_s3_max = pd.read_csv('s3_timeSnitt_2018_max.csv')
df_s3_max = df_1103_max.iloc[196:, [36, 34,
                                    68, 70, 72, 74, 76, 78, 86, 88, 90,
                                    92, 94, 96, 104, 106, 108, 110, 112,
                                    114, 122, 124, 126, 128, 130, 132,
                                    80, 82, 84, 134, 136, 138]]
df_s3_min = pd.read_csv('s3_timeSnitt_2018_min.csv')
df_s3_min = df_1103_min.iloc[196:, [36, 34,
                                    68, 70, 72, 74, 76, 78, 86, 88, 90,
                                    92, 94, 96, 104, 106, 108, 110, 112,
                                    114, 122, 124, 126, 128, 130, 132,
                                    80, 82, 84, 134, 136, 138]]

#Concat min max
df_s3 = pd.concat([df_s3_min, df_s3_max], axis = 1)
#Time on index
date_rng = pd.date_range('01/09/2018', '01/01/2019', freq = 'H')
date_rng = date_rng.delete(8568)
df_s3['dateTime'] = date_rng
df_s3.set_index(pd.to_datetime(df_1103['dateTime']), inplace=True, drop=True)
df_s3 = df_s3.drop(['dateTime'], axis = 1)


#Append wind and rain
windRain = pd.read_csv('windRain2018.csv')
windRain = windRain.set_index(pd.to_datetime(windRain['referenceTime']))
windRain = windRain.drop('referenceTime', axis = 1)
windRain = windRain.fillna(windRain.mean())
windRain = windRain.iloc[191:,:]
df_s3 = pd.concat([df_s3, windRain], axis = 1)
df_s3 = df_s3.fillna(windRain.mean())

#remove repeating values
df_s3_1 = df_s3.iloc[:3264,:]
df_s3_2 = df_s3.iloc[5304:,:]
df_s3_stripped = pd.concat([df_s3_1, df_s3_2])
```

```
#import S1 and S2
#importing the datasets and extracting the features of interest
df_2710_s1 = pd.read_csv('Data_2710_s1.csv', sep=',', encoding='utf-8')
df_2710_s1  = df_2710_0912.iloc[4:, [6, 60,
                                       42, 44, 46, 48, 50, 52, 68, 70, 72,
                                       74, 76, 78, 92, 94, 96, 98, 100, 102,
                                       116, 118, 120, 122, 124, 126,
                                       54, 56, 58, 128, 130, 132]]

df_0101_s1 = pd.read_csv('Data_0101_s1.csv', sep=',', encoding='utf-8')
df_0101_s1  = df_0101_0912.iloc[4:, [6, 60,
                                       42, 44, 46, 48, 50, 52, 68, 70, 72,
                                       4, 76, 78, 92, 94, 96, 98, 100, 102,
                                       116, 118, 120, 122, 124, 126,
                                       54, 56, 58, 128, 130, 132]]

df_2710_s2 = pd.read_csv('Data_2710_s2.csv', sep=',', encoding='utf-8')
df_2710_s2 = df_2710_6358.iloc[4:, [6, 34,
                                      16, 18, 20, 22, 24, 26, 44, 46, 48,
                                      50, 52, 54, 68, 70, 72, 74, 76, 78,
                                      92, 94, 96, 98, 100, 102,
                                      28, 30, 32, 104, 106, 108]]

df_0101_s2 = pd.read_csv('Data_0101_s2.csv', sep=',', encoding='utf-8')
df_0101_s2 = df_0101_6358.iloc[4:, [6, 34,
                                      16, 18, 20, 22, 24, 26, 44, 46, 48,
                                      50, 52, 54, 68, 70, 72, 74, 76, 78,
                                      92, 94, 96, 98, 100, 102,
                                      28, 30, 32, 104, 106, 108]]

#Weatherdata from 27/10 and 01/01
windRain0101 = pd.read_csv('windRain010119.csv', sep=',', encoding='utf-8')
windRain0101.index = windRain0101['referenceTime']
windRain0101 = windRain0101.drop('referenceTime', axis = 'columns')

windRain2710 = pd.read_csv('windRain271018.csv',
                           delimiter = ',', encoding='utf-8')
windRain2710.index = windRain2710['referenceTime']
windRain2710 = windRain2710.drop('referenceTime', axis = 1)
```

```python
#fixing the dataframe
def reform(df, date = ['10/27/2018', '10/28/2018'], freq = 'H'):
    date_rng = pd.date_range(date[0], date[1], freq='S')
    date_rng = date_rng.delete(86400)

    df['datetime'] = date_rng
    df.set_index(pd.to_datetime(df['datetime']), inplace=True, drop=True)
    df = df.drop(['datetime'], axis = 1)
    df = df.apply(pd.to_numeric) #all columns of DataFrame to numerics

    #Resampling to hourly samples
    #meanHour = df.resample('1H').mean()
    maxHour = df.resample(freq).max()
    minHour = df.resample(freq).min()

    #Create a frame containing min and max values
    dfHour = [minHour, maxHour]
    dfHour = pd.concat(dfHour, axis = 1)
    dfHour.columns = ['Outdoor temp min',
                      'Indoor temp min',
                      '0.4 T1-NRG   SPENNING L1-L2 min',
                      '0.4 T1-NRG   SPENNING L1-N min',
                      '0.4 T1-NRG   SPENNING L2-L3 min',
                                    .
                                    .
                                    .
                      '22 F53 current L1 max',
                      '22 F53 current L2 max',
                      '22 F53 current L3 max']

    if (date == ['10/27/2018', '10/28/2018']):
        dfHour = pd.merge(dfHour, windRain2710, how='left', left_index = True,
                          right_index = True).fillna(windRain2710.mean())
    elif (date == ['01/01/2019', '01/02/2019']):
        dfHour = pd.merge(dfHour, windRain0101, how='left', left_index = True,
                          right_index = True).fillna(windRain0101.mean())
    dfHour.to_csv(path_or_buf = 'ReformedData.csv')
    return dfHour

#Reform imported minute-files
df_2710_0912_hours = reform(df_2710_0912,
```

```
                            date = ['10/27/2018', '10/28/2018'], freq = 'H')
df_2710_6358_hours = reform(df_2710_6358,
                            date = ['10/27/2018', '10/28/2018'], freq = 'H')
df_0101_0912_hours = reform(df_0101_0912,
                            date = ['01/01/2019', '01/02/2019'], freq = 'H')
df_0101_6358_hours = reform(df_0101_6358,
                            date = ['01/01/2019', '01/02/2019'], freq = 'H')

df_2710_0912_minutes = reform(df_2710_0912,
                              date = ['10/27/2018', '10/28/2018'], freq = 'T')
df_2710_6358_minutes = reform(df_2710_6358,
                              date = ['10/27/2018', '10/28/2018'], freq = 'T')
df_0101_0912_minutes = reform(df_0101_0912,
                              date = ['01/01/2019', '01/02/2019'], freq = 'T')
df_0101_6358_minutes = reform(df_0101_6358,
                              date = ['01/01/2019', '01/02/2019'], freq = 'T')
```

A.2 Preprocessing and Fit the three models

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from datetime import datetime
from collections import Counter
from sklearn.preprocessing import StandardScaler
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn import metrics

dataset_substation_s3_max = pd.read_csv('/CSVfiles/1103_timeSnitt_2018_max.csv'
                                        , sep=',', encoding='utf-8')
N = dataset_substation_s3.iloc[196:, [34, 36, 70, 74, 78, 88, 92, 96,
                                      106, 110, 114, 124, 128, 132]]

#Label the data
def f(row):
    if (row['F51 L1-N min']<1 or
        row['F52 L1-N min']<1 or
        row['F53 L1-N min']< 1):
        val = -1  #fault = -1
    else:
        val = 1 #healthy = 1
    return val

df_X['label'] = df_X.apply(f, axis=1)


df_X['label'] = df_X['label'].shift(-1) #Shifting the labels up
df_X = df_X.dropna() #Dropping bottom row

df_X.to_csv(path_or_buf = 'StromOgLabels.csv')

#Split labels from features
X = df_1103_weather.iloc[:,:-1]
Y = df_1103_weather.iloc[:,-1]

# Feature Scaling
sc_X = StandardScaler()
```

```
X = sc_X.fit_transform(X) #returns an array
X = pd.DataFrame(data = X)

# Splitting the dataset into the Training set and Test set
splitPoint = 7104 #index 7104 = 1 November(00:00)
X_train = X.iloc[:splitPoint, : ]
X_test = X.iloc[splitPoint:, : ]
y_train = Y.iloc[:splitPoint]
y_test = Y.iloc[splitPoint:]


#Random Forest Classification
#Fitting Random Forest Regression to the dataset
regressor = RandomForestClassifier(n_estimators = 100, random_state = 0)
regressor.fit(X_train, y_train)

#Make predictions on the test set
predictions = regressor.predict(X_test)
y_test_float = np.float64(y_test).flatten()




# ================================================================================
# ONE class SVM
# ================================================================================

df_2710_0912_labels_run = repeatingSVM(df_1103_stripped,
                                       df_2710_0912_hours,
                                       df_2710_0912_labels)
df_2710_6358_labels_run = repeatingSVM(df_1103_stripped,
                                       df_2710_6358_hours,
                                       df_2710_6358_labels)
df_0101_0912_labels_run = repeatingSVM(df_1103_stripped,
                                       df_0101_0912_hours,
                                       df_0101_0912_labels)
df_0101_6358_labels_run = repeatingSVM(df_1103_stripped,
                                       df_0101_6358_hours,
                                       df_0101_6358_labels)
```

```
def repeatingSVM(df_train, df_test, df_labels):
    df_test_labels = pd.DataFrame(pd.np.empty((24, 0)))
    df_test_labels['True Label'] = df_test_labels.join(df_labels)
    sc_X = StandardScaler()
    df_train_sc = sc_X.fit_transform(df_train)
    df_train_sc = pd.DataFrame (data = df_train_sc)
    df_test_sc = sc_X.fit_transform(df_test)
    df_test_sc = pd.DataFrame (data = df_test_sc)

    for i in range(0,5):
        X_train, X_test = train_test_split(df_train_sc)
        #X_train=random75% of df_train
        clf = svm.OneClassSVM (nu = 0.05, gamma = 'scale')
        clf.fit(X_train)
        df_pred = clf.predict(df_test_sc)
        df_pred = pd.DataFrame(data = df_pred)
        df_test_labels[str(i)] = df_pred[0]
    return df_test_labels

# ==============================================================================
# Below - Second unsupervised approach: Minute measurements
# ==============================================================================

#splitting to train and test
df_2710_0912_t_train = df_2710_0912_minutes.iloc[:600,:]
df_2710_0912_t_train = df_2710_0912
                            _minutes.append(df_2710_0912_minutes.iloc[1080:,:])
df_2710_0912_t_test = df_2710_0912_minutes.iloc[600:1080,:]

df_2710_6358_t_train = df_2710_6358_minutes.iloc[:480,:]
df_2710_6358_t_train = df_2710_6358
                            _minutes.append(df_2710_6358_minutes.iloc[900:,:])
df_2710_6358_t_test = df_2710_6358_minutes.iloc[480:900,:]

df_0101_0912_t_train = df_0101_0912_minutes.iloc[:210,:]
df_0101_0912_t_train = df_0101_0912
                            _minutes.append(df_0101_0912_minutes.iloc[1080:,:])
df_0101_0912_t_test = df_0101_0912_minutes.iloc[210:1080,:]

df_0101_6358_t_train = df_0101_6358_minutes.iloc[:480,:]
```

```
df_0101_6358_t_train = df_0101_6358
                            _minutes.append(df_0101_6358_minutes.iloc[960:,:])
df_0101_6358_t_test = df_0101_6358_minutes.iloc[480:960,:]


def OCSVM(df_train, df_test):
    #Scale
    sc_X = StandardScaler()
    df_train_sc = sc_X.fit_transform(df_train)
    df_train_sc = pd.DataFrame(data = df_train_sc)
    df_test_sc = sc_X.fit_transform(df_test)
    df_test_sc = pd.DataFrame(data = df_test_sc)
    #Create classifier
    clf = svm.OneClassSVM (nu = 0.05, gamma = 'scale')
    clf.fit(df_train_sc)
    #Make predictions
    df_pred = clf.predict(df_test_sc)
    df_pred = pd.DataFrame(data = df_pred)
    df_pred.index = df_test.index
    pred_score = clf.score_samples(df_test_sc)
    train_score = clf.score_samples(df_train_sc)
    dec_func = clf.decision_function(df_train_sc)

    return df_pred, pred_score, train_score, dec_func


df_test = df_0101_0912_minutes.iloc[480:960,:]
df_train = df_0101_0912_minutes.iloc[:480,:]
df_train = df_train.append(df_0101_0912_minutes.iloc[960:,:])

df_pred, pred_score, train_score, dec_func = OCSVM(df_train, df_test)
```

Fredrik Heistad, William Andreas Kristensen

NTNU
Norwegian University of
Science and Technology