Halvor Kvernes Meen and Joachim Jahr

Cluster Analysis of Faults in the Norwegian Power Grid

Specialization Project, Autumn 2019

Supervisor: Helge Langseth

Artificial Intelligence Group
Department of Computer and Information Science
Faculty of Information Technology, Mathematics and Electrical Engineering
Norwegian University of Science and Technology (NTNU)

Abstract

The modern society has grown dependant on electricity and as such the power grid has become a crucial part of our infrastructure. Providing a stable power distribution network is of utter importance, ensuring that both industry and households have a predictable source of energy. With the advances of machine learning and storage capacities of big data, there have emerged a wish to predict faults on the degrading power grid in order to assure stability for the users.

In this report we explore whether there are underlying structures in data obtained from the Norwegian Power grid, and if it is possible use these structures to differentiate between faults and normative operations prior to the fault occurring. To do this we present some suitable methods of clustering and dimensionality reduction for analysis of the data. We then use these clustering methods on the data and inspect whether there are distinctive clusters of either faults or normative behaviours. We discover that some faults differentiate considerably from normative behaviour, but most faults do not act different from normative prior to the fault occurring when inspecting the mean, maximum and standard deviation of Fourier harmonics of the current.

Keywords Norwegian Power Grid, Clustering, Machine Learning

Acknowledgements

First of all we would like to thank our supervisor Helge Langseth for sharing his knowledge with us and for his support throughout this long process of writing this report. His weekly feedback and meetings have been of great help and motivation.

We would also like to thank everyone at SINTEF that supported us, Christian Andresen, Gjert Hovland Rosenlund, Kristian Wang Høiem and Bendik Torsæter. Thank you for clarifying the project and for providing us with the resources we needed.

Table of Contents

List of Tables

List of Figures

Abbreviations

AC	=	Alternating Current
A-HA	=	Automatisk Hendelsesanalyse
AUC	=	Area Under the Curve
A	=	Current
CNN	=	Convolutional Neural Network
CPU	=	Central Processing Unit
DC	=	Direct Current
DDG	=	Dynamic Dataset Generator
EM	=	Expectation Maximization
FN	=	False Negative
FP	=	False Positive
GPU	=	Graphics Processing Unit
GMM	=	Gaussian Mixture Model
ILE	=	Ikke Levert Energi
MCC	=	Matthews Correlation Coefficient
P	=	Power
PCA	=	Principal Component Analysis
PMU	=	Phasor Measurement Unit
PQA	=	Power Quality Analyzers
RMS	=	Root Mean Square
ROC	=	Receiver Operating Characteristic
SNR	=	Signal-to-Noise Ratio
TN	=	True Negative
TP	=	True Positive
t-SNE	=	t-distributed Stochastic Neighbor Embedding
V	=	Voltage

Chapter 1

Introduction

This report is a part of the EarlyWarn project. The main purpose of EarlyWarn is to develop surveillance systems that discover and identify faults and disturbances in the Norwegian power grid. EarlyWarn is explained in more detail in Chapter 3.

1.1 Motivation

The modern society has grown dependant on electricity and as such the power grid has become a crucial part of our infrastructure. This dependency has grown stronger and stronger since Edison invented the light bulb in the late 1800's until today where we cannot imagine a day without our smartphones. The power grid is not only important for the daily life of people, but also for businesses and for the government to function properly. This has put very high quality and reliance expectations on the power grid and on the workers that operate it. This is especially true for Norway and other northern countries as we rely on electricity to stay warm in the winter. The Norwegian power grid amounts to more than 130 000 km of transmission lines. Even though it already has been extensively developed, many billions are invested annually for improvement and further expansion. The Norwegian power grid has been subject to heavy investments since the mid 2000's [Sentralbyrå, 2016]. In 2019 all the investments total to about 40 billion NOK which was a small downfall from 2018, but seen in a historic perspective, it is still a considerable sum [Sentralbyrå, 2019a]. Number of investments for the last years can be seen in Figure 1.1. The Norwegian industry has also had a steady increase in energy consumption over the last years [Sentralbyrå, 2019b].

The Nordic power grids are currently undergoing the most significant changes in more than 20 years [e24, 2018]. These changes are largely motivated by a focus on the climate

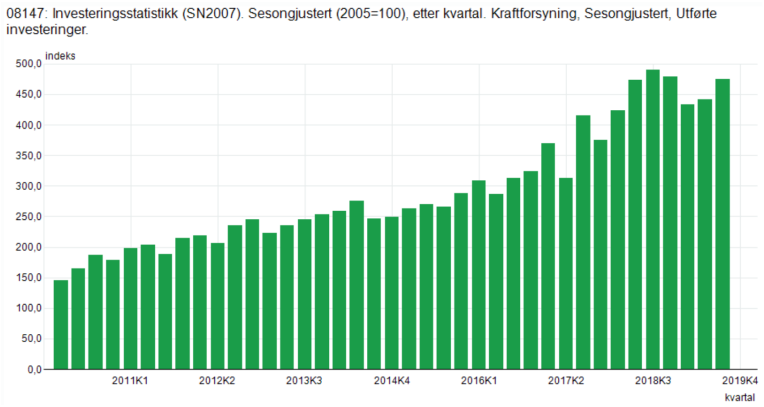


Figure 1.1: Statistics for the period 2011-2019 showing the number of investments adjusted after seasons (Statistisk Sentralbyrå).

and being more Eco-friendly. We can expect to see more use of smart power measurement devices and new technologies allowing for automatic power adjustments.

With access to data gathered from sensors placed all around the grid, and by advancement in machine learning technologies in combination with domain knowledge of faults and disturbances in the power grid, EarlyWarn aims to improve the overall reliability of the power grid by being able to predict and hopefully being able to prevent faults before they occur. By being able to take preemptive measures against possible faults, the cost of maintenance and repairs might be reduced drastically.

1.2 Research Questions

The main goal of this report is to explore to which extent the data obtained through the EarlyWarn project have intrinsic structures, and if so, if these structures can be used to differentiate between faults and non-faults. It would also be interesting to find out if some if these structures might be interpretable to humans, such that operators can take advantage of them during analysis. In order to find these structures, clustering will be performed on the data. This is to see if these unsupervised methods can locate substantial differences between the sampled data corresponding to the differences between labels. To help explore these goals, the following research questions have been formulated:

- **RQ1:** What is the impact of dimensionality reduction, and how does it affect the clustering results?
- **RQ2:** To what extent do there exist underlying structures in the data?
- **RQ3:** How can the structures be used to differentiate between faults and non-faults?

Chapter 2

Background - Power Grids

In this chapter we will briefly explain the fundamental concepts of the power grid. We will also take a look at which faults and disturbances that can occur, and the circumstances that causes them.

2.1 Introduction

A power grid (or electrical grid) has the responsibility of transferring electric power from a producer to a consumer, and usually consists of; generating stations (producers), substations (transforms the **voltage**), transmission lines (transfers the **power**) and consumers. We will from here on refer to electrical power as just power. Another term that is highly related to both power and voltage is **current**. Power, current and voltage are defined as follows:

- **Power** (P) is the rate of energy consumption per time unit and is measured in units of watts (joule per second).
- **Current** (A) is the rate of flow of electric charge past a point and is measured in units of amperes (coulomb per second).
- **Voltage** (V) is the difference in potential electric energy between two points and is measured in units of volts (joule per coulomb).

The power grid of interest is the Norwegian power grid which is making sure that all citizens and other consumers have access to the electricity they need. We will from here on

refer to the Norwegian power grid as just ”the power grid”. The power grid is traditionally divided into three nets:

- The **transmission net** which represents the highest voltage levels (normally between 300kV to 420kV) and transmits power over huge distances throughout the country. This also includes connections to neighbouring countries. It amounts to 11 000 km of transmission lines.
- The **regional net** which represents the middle voltage levels (normally between 33kV and 132kV) and is a middle layer between the transmission net and the distribution net. It amounts to 19 000 km of transmission lines.
- The **distribution net** which represents the lowest voltage levels (up to 22kV) and is the final link that transmits power to the end consumer. It amounts to 100 000 km of transmission lines. The distribution net is further separated into a high voltage part and a low voltage part, where the separation is at 1kV and the low voltage part usually is either 400V or 230V for normal consumption.

The three nets together amounts to a total of 130 000 km of transmission lines where the distribution net has the biggest contribution. All the nets are different in nature and therefore has different challenges that must be addressed. Unique of these three is the vast distribution net which with its huge size and complex structure makes it more prone to faults and disturbances which we will closer into later.

2.2 Fundamentals

2.2.1 Direct- and Alternating Currents

There are two types of currents; direct currents (DC) and alternating currents (AC). Direct current is the most basic one where the current is constantly flowing in one direction. Alternating current is, as revealed from the name, alternating the direction of the current flow (See Figure 2.1). This means that while DC is a steady source of power, AC provides a flow of power that is in varying in strength. How fast the direction of the flow is alternated, the frequency, is measured in units of hertz (Hz, switches per second). The frequency is dependant on the country and is usually either 50Hz or 60Hz. The frequency in Norway is 50Hz.

There are several benefits with using AC that makes it the preferred choice over DC when it comes to power grids, but the main reason is that the voltage can be transformed to higher or lower voltage levels depending on the usage. This is crucial as high voltage levels are much more efficient when transferring power over big distances while the end consumers only need a fraction of those voltage levels. High voltages are more efficient because it requires less current which in turn reduces the overall power loss.

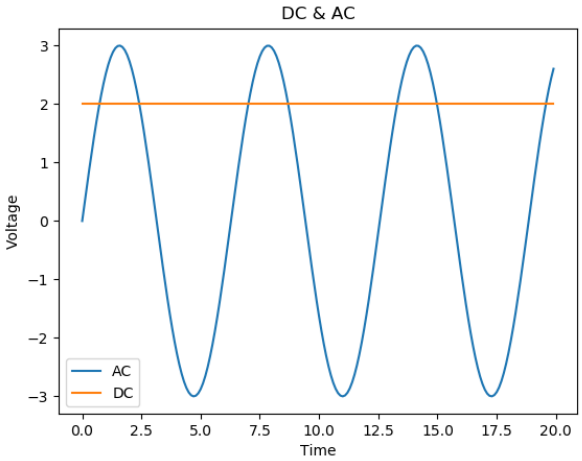


Figure 2.1: Example of a direct- and an alternating current.

2.2.2 Mathematical Representations

The AC voltage v and current i can be described mathematically as a function of time t:

$$v(t) = V_m \cos(\omega t + \varphi_v) \tag{2.1}$$

$$i(t) = I_m \cos(\omega t + \varphi_i)$$

where V_m and I_m is the maximum amplitude for voltage and current respectively (peak voltage and peak current), ω is the angular frequency[1] measured in units of radians per second, and φ_v and φ_i are the phase angles between the voltage and the current.

A popular way of representing a sinusoidal wave is a concept called a *phasor*. A phasor is simply put a vector representing the wave with a rotating motion in the complex plane. To be able to represent a sinusoidal it is crucial that the amplitude, angular frequency and phase angle are invariant to time. This is because the length of the vector is constant and will be equal to the maximum amplitude. (See Figure 2.2 for visualization).

By using Euler's formula:

$$e^{ix} = \cos x + i \sin x \tag{2.2}$$

where e is Euler's number and i is the imaginary unit, we can rewrite Equation 2.1 to

[1] $\omega = 2\pi f$ where f is the cyclic frequency measured in the unit of hertz.

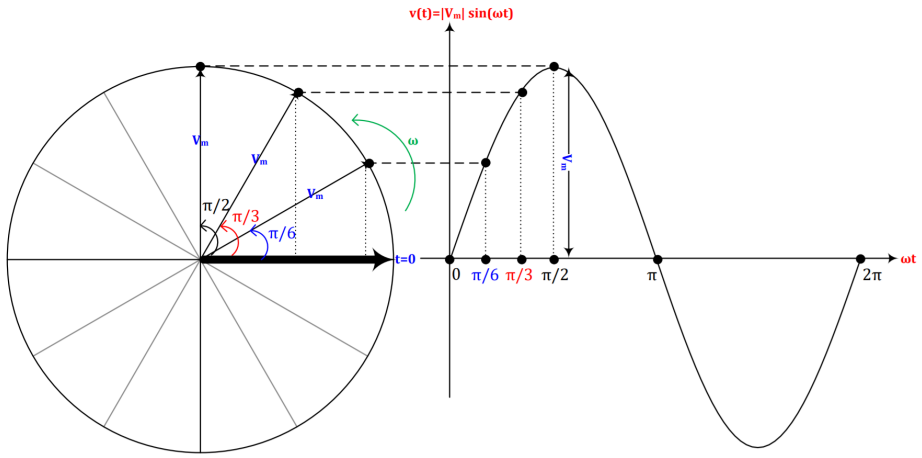


Figure 2.2: The relationship between a phasor and sinusoidal wave [Vadlamudi, 2018].

[Vadlamudi, 2018]:

$$\begin{aligned} v(t) &= V_m \cos(\omega t + \varphi_v) \\ &= Re(V_m e^{i(\omega t + \varphi_v)}) \\ &= Re(V_m e^{i\varphi_v} e^{i\omega t}) \end{aligned} \tag{2.3}$$

where *Re* is the real part of the complex equation. To find the vector for the phasor representation we rewrite Equation 2.3 to:

$$v(t) = Re(\mathbf{V} e^{i\omega t})$$

where $\mathbf{V}$ is the phasor representation defined as $\mathbf{V} = V_m e^{i\varphi_v}$.

2.2.3 Three Phase Power

As earlier explained, AC is not a constant power source. It varies in strength as it goes from the positive voltage peak V_m where it gives maximum power, and gets weaker as it goes towards zero. It then gets stronger again until it reaches the negative voltage peak where it also gives maximum power (in the opposite direction). This results in an uneven flow of power which can cause problems such as flickering lights. By introducing two more phases the instantaneous power will be constant, meaning that even though the three phases on their own will vary, combined they will provide a constant source of power (See Figure 2.3).

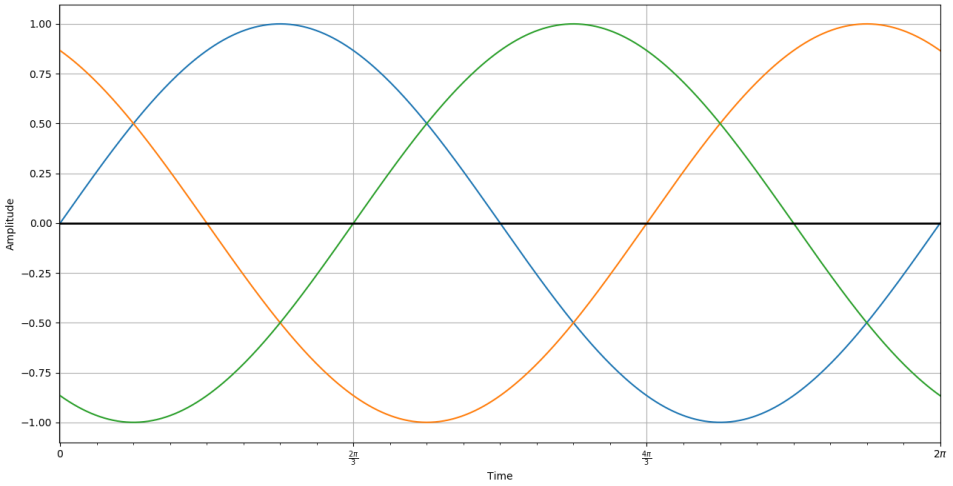


Figure 2.3: Sinusoidal wave representation of three phase power with $\frac{2}{3}\pi$ radians as phase offset.

To build a three phase generator three coils are placed $\frac{2}{3}\pi$ radians (120°) apart (See 2.4 around a rotating magnet. The three phases all have the same magnitude and angular frequency for both voltages and currents. There are numerous advantages with using a three phase power system [Vadlamudi, 2018]; Can transmit more power for same amount of wire, can start more easily, power transfer is constant which reduces generator and motor vibrations. There is also disadvantages as there are triple the amount of phases, which results in a greater risk that one of them will fail and cripple the system.

2.2.4 Root Mean Square (RMS)

As the sinusoidal wave representation of an alternating current has different values dependant on the time, it would be nice with a single value independent of time to describe the voltage. A common measurement is the average value. This is not helpful when looking directly at the sinusoidal waves as they half the time are positive and rest of the time are negative, which results in an average of zero (assuming you calculate over a period). **RMS** avoids this problem by taking the square of the wave resulting in only positive values. RMS is defined as:

$$V_{RMS} = \sqrt{\frac{1}{T_2 - T_1.} \int_{T_1}^{T_2} v(t)^2 dt}$$

where *v(t)* is a sinusoidal function with period T[2]. The RMS can be further simplified by substituting in the function for *v(t)* from Equation 2.1 (can ignore the phase angle φ_v) and

[2]Here the RMS is defined in respect to the voltage, but can equivalently be defined in respect to current by replacing V_{RMS} with I_{RMS} and *v(t)* with *i(t)*

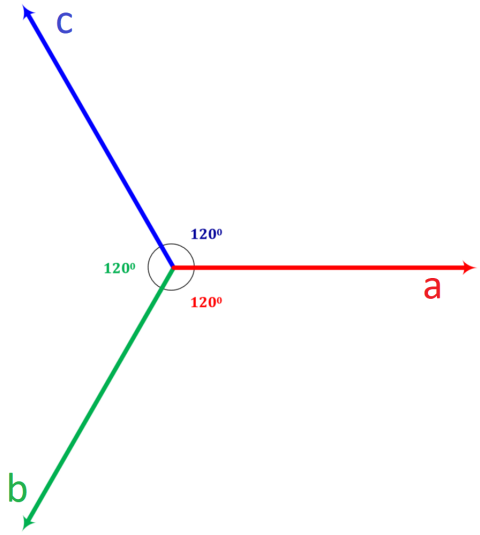


Figure 2.4: Phasor diagram representation of three phase power, phases a, b and c, with $\frac{2}{3}\pi$ radians (120°) as phase offset.

by using the trigonometric identity $cos^2(x) = \frac{1}{2}(1 + cos(2x))$:

$$\begin{aligned} V_{RMS} &= \sqrt{\frac{1}{T_2 - T_1}\int_{T_1}^{T_2} v(t)^2 dt} \\ &= \sqrt{\frac{1}{T_2 - T_1}\int_{T_1}^{T_2} V_m^2 \cos^2(\omega t) dt} \\ &= V_m\sqrt{\frac{1}{T_2 - T_1}\int_{T_1}^{T_2} \frac{1}{2}(1 + \cos(2\omega t)) dt} \\ &= V_m\sqrt{\frac{1}{T_2 - T_1}\left[\frac{t}{2} + \frac{\sin(2\omega t)}{4}\right]_{T_1}^{T_2}} \end{aligned} \tag{2.4}$$

where T_1 and T_2 are the start and ending periods respectively, such that the interval is one complete cycle. This results in the *sin* terms in Equation 2.4 cancelling out, leaving:

$$\begin{aligned} V_{RMS} &= V_m\sqrt{\frac{1}{T_2 - T_1}\frac{T_2 - T_1}{2}} \\ &= \frac{V_m}{\sqrt{2}} \end{aligned}$$

Subsequently RMS gives the time-averaged power that the AC delivers which also is equal to the power delivered by a DC voltage with matching value. RMS is very useful to observe

in regards to faults and disturbances. Deviations in the RMS value imply that there might be an error within the system. However, deviations in RMS alone is not always enough to determine if there has been an error and might sometimes require further investigation.

2.2.5 Fourier Transform

The Fourier transform is a function that decomposes a waveform into its fundamental frequencies, and by so transforming it from the time domain to the frequency domain. The Fourier transform $\hat{f}$ can be defined as:

$$\hat{f}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-2\pi i t \omega} dt$$

where f is the input waveform, ω is the frequency and t is the time. The original waveform f can be reconstructed by doing the inverse transform on $\hat{f}$:

$$f(t) = \int_{-\infty}^{\infty} \hat{f}(\omega) e^{2\pi i t \omega} d\omega$$

As previously defined, the Fourier transform is performed on a continuous function (thereof the integration), but in a more realistic setting we do not have the capacity/ability to sample a function for all values of time. Instead we sample the function with a certain time interval resulting in discrete samples in contrast to the whole continuous function. We further define the Discrete Fourier transform X_k of a series x_n with N samples as:

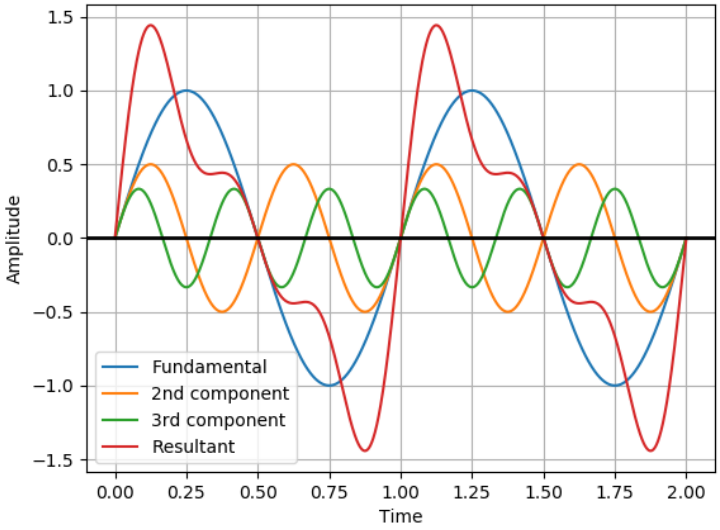
$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} k n}$$

where n is a natural number. As with the continuous transform we can also find the inverse:

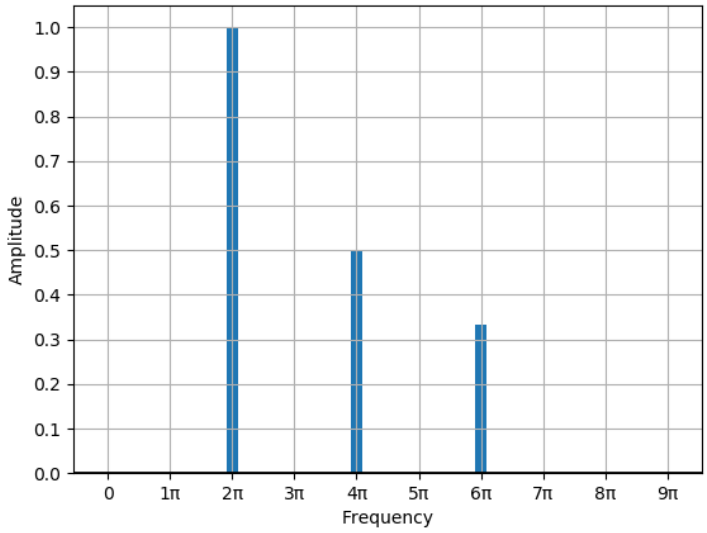
$$x_n = \frac{1}{N} \sum_{k=0}^{N-1} X_k e^{\frac{2\pi i}{N} k n}$$

By using Euler's formula (Equation 2.2) with $x = \frac{2\pi}{N} kn$ we can rewrite the Discrete Fourier transform as:

$$\begin{aligned} X_k &= \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} k n} \\ &= \sum_{n=0}^{N-1} x_n (\cos(\frac{2\pi}{N} kn) - i \sin \frac{2\pi}{N} kn) \\ &= \sum_{n=0}^{N-1} x_n \cos(\frac{2\pi}{N} kn) - i \sum_{n=0}^{N-1} x_n \sin(\frac{2\pi}{N} kn) \end{aligned}$$



(a)



(b)

Figure 2.5: (a) shows a sinusoidal function with its 3 components. (b) shows the coefficients of its Discrete Fourier transform.

A visual representation of a sinusoidal function and its Fourier transform can be seen in Figure 2.5. Figure 2.5(a) displays a sinusoidal function with components $\sin 2\pi x$ (fundamental), $\frac{1}{2}\sin 4\pi x$ (2nd component) and $\frac{1}{3}\sin 6\pi x$ (3rd component), with frequency 2π, 4π and 6π, and amplitude 1, $\frac{1}{2}$ and $\frac{1}{3}$ respectively. Figure 2.5(b) shows the Fourier coefficients, the frequencies, with the belonging amplitudes.

2.2.6 Harmonics

In regards to electric power systems, a harmonic is a multiple of the fundamental frequency of the system. They appear as both voltage and current. Harmonics are generally unwanted as they distort the pure sinusoidal wave of the system, and can cause problems such as increased heat dissipation.

More formally, if we have a fundamental frequency (also referred to as the 1st harmonic) of the system *f*, the harmonics have a frequency of nf where *n* is a natural number (See Figure 2.6 for a visual representation).

The distorted sinusoidal can be decomposed by using the Discrete Fourier transform, giving a infinite series representation of harmonic components:

$$v(t) = V_{avg} + \sum_{k=1}^{\infty} V_k \sin\left(k\omega t + \varphi\right)$$

where V_{avg} is the average amplitude (also often referred to as the DC value) and V_k is the amplitude of the *k*th harmonic.

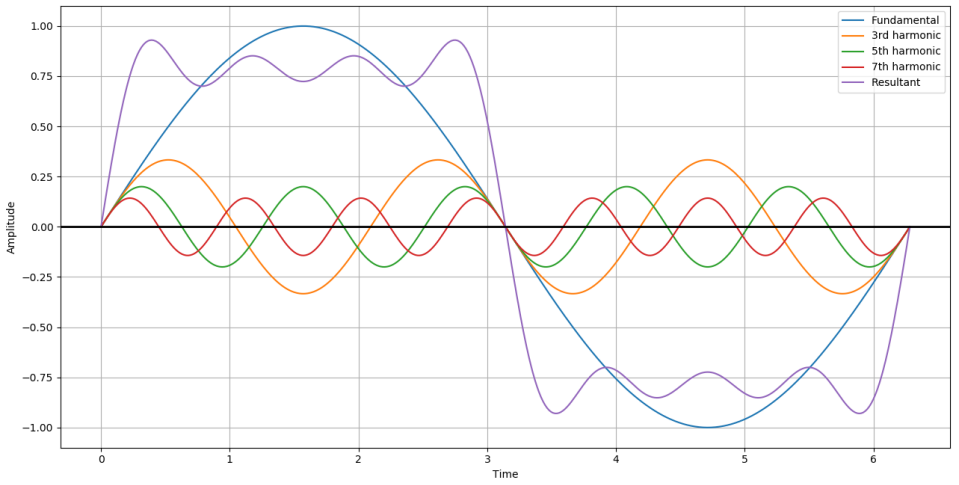


Figure 2.6: Resultant of the 1st, 3rd, 5th and 7th harmonic.

2.3 Faults and Disturbances

There are three nationwide statistics compiled annually regarding the Norwegian power grid:

- *Avbrotsstatistikk* [Norges vassdrags-og energidirektorat, 2019], which is a statistic of interrupts reported by multiple participating companies and end users. For the year 2018 it was compiled on the basis of data from 111 reporting companies and approx. 3.11 million end users. The total energy delivered to the end users was approx. 121 TWh.
- *"Driftsforstyrrelser, feil og planlagte utkoplinger i 1-22 kV-nettet* [Statnett, 2019a], which provides an overview of scheduled downtime due to maintenance, operational faults and interruptions in the 1-22 kV grid (i.e. the distribution net).
- *Driftsforstyrrelser og feil i 33-420 kV-nettet (inkl. driftsforstyrrelser pga. produksjonsanlegg)* [Statnett, 2019b], which provides an overview of scheduled downtime due to maintenance, operational faults and interruptions in the 33-420 kV grid (i.e. the transmission- and regional net).

According to [Norges vassdrags-og energidirektorat, 2019] power that could not be delivered due to interruptions amounted to 0.017% of the total delivered energy in 2018. This means the power delivery reliability was 99,983%. Furthermore, according to [Statnett, 2019a] and [Statnett, 2019b] there were 10798 operational faults on the distribution net, which was a lot higher than normal, but only 740 operational faults on the transmission- and regional net, which was very low compared to previous years. As noted earlier, there are overwhelmingly more faults on the distribution net as it contains most of the transmission lines as well as it has a complex structure.

Faults can range from natural occurrences such as a tree falling on the line or icing in the winter, to wear and tear of equipment. *Statnett*[3] has made a categorization utilized in the annual reports and can be viewed in Figure 2.7 in context of operational faults and in Figure 2.8 in context of undelivered power (ILE[4]). As can be seen in the figures, *surroundings* are the biggest cause of both operational faults and ILE. The surroundings was further categorized into subcategories as can be seen in Figure 2.9 and Figure 2.10. Apparent from these figures is that only thunderstorms were a consistent cause in both 2018 and the mean of previous years, while vegetation was the biggest factor in 2018 and wind for the previous years. Surprisingly wind is the dominant cause of ILE for previous years while vegetation was the dominating cause for 2018. The reason why wind has been the dominating cause of ILE even though thunderstorms caused most operational faults can be explained by that faults caused by wind have done more damage in comparison, resulting in more severe faults.

[3] *Statnett* is a Norwegian state owned enterprise responsible for owning, operating and constructing the power grid in Norway.

[4] **I**kke **L**evert **E**nergi in Norwegian.

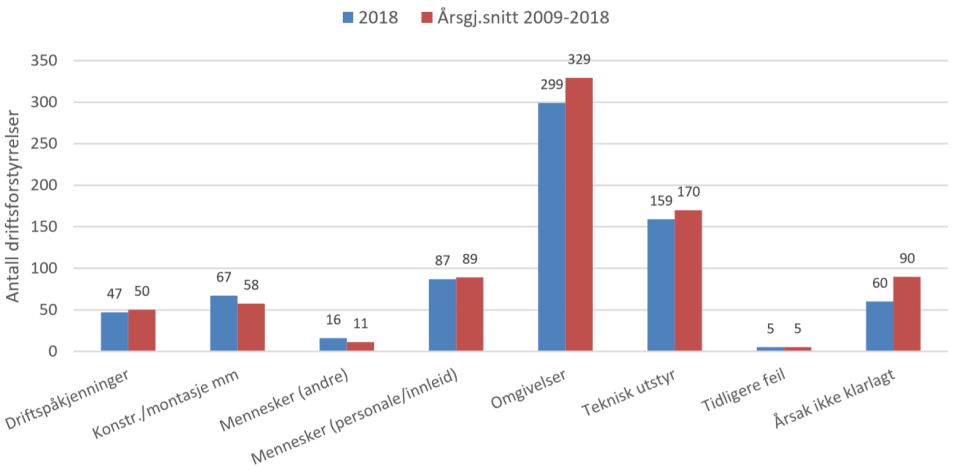


Figure 2.7: Overview of operational faults on the transmission- and regional net and their causes.

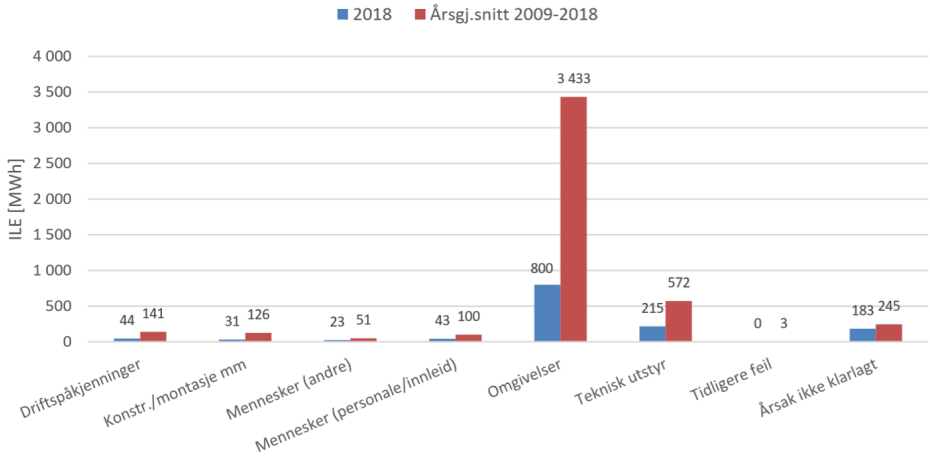


Figure 2.8: Overview of ILE on the transmission- and regional net and their causes.

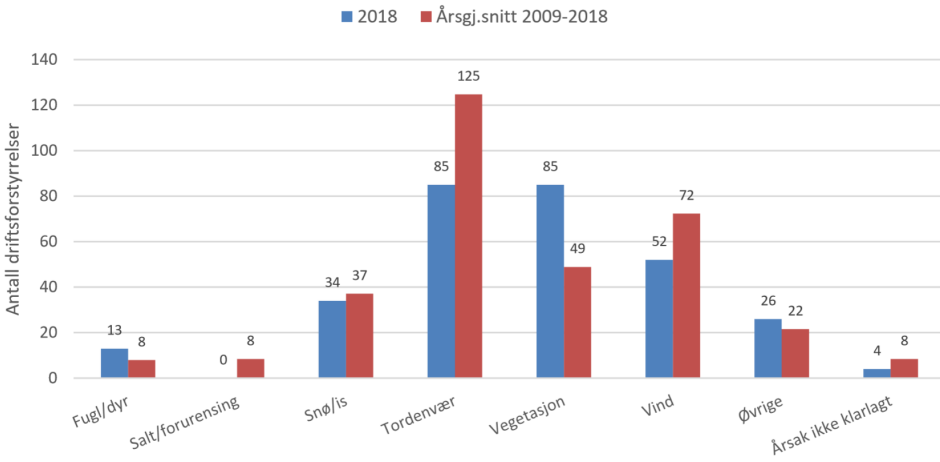


Figure 2.9: Overview of operational faults on the transmission- and regional net caused by surroundings.

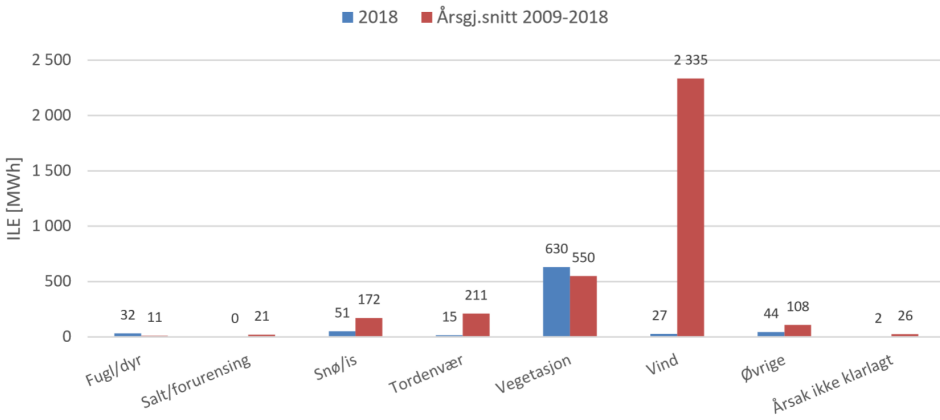


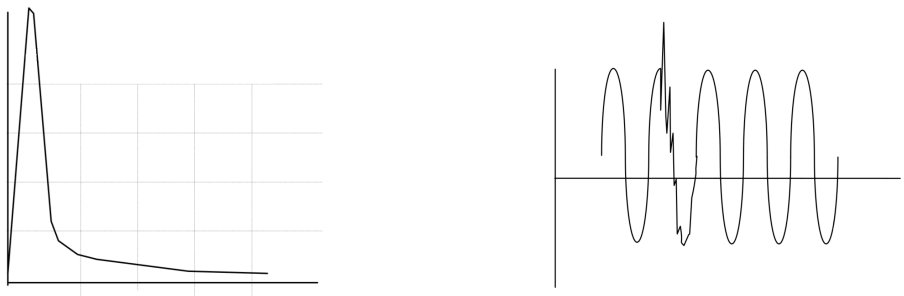
Figure 2.10: Overview of ILE on the transmission- and regional net caused by surroundings.

We are mostly interested in faults that have the possibility of being recognized by looking at disturbances in the power signal. Faults like a tree falling on the transmission line or a bird causing a shorting are therefore out of the scope of this report. So far we have only discussed causes of faults in the big picture. We will now take a closer look at faults in respect to the power signal. [Seymour, 2001] organized power disturbances into seven different categories based on the shape of the wave:

1. **Transients**
2. **Interruptions**
3. **Sag / Undervoltage**
4. **Swell / Overvoltage**
5. **Waveform distortion**
6. **Voltage fluctuations**
7. **Frequency variations**

Transients, which were referred to as the potentially most damaging type of power disturbance, can further be divided into two subcategories (See Figure 2.11); impulsive and oscillatory transients. Impulsive transients is the most common type of power surge/spike and involves a sudden increase or decrease of the voltage/current level. It is usually spans a very short time interval. Causes include lightning, grounding failure and equipment faults to name a few. Oscillatory transients causes disturbances in the power signal, making the signal jump between low and high values resulting in a oscillating motion. Often caused by a sudden loss of a load.

Interruptions are defined as a complete loss of voltage/current (See Figure 2.12) and can further be divided into four subcategories in respect to the durations; instantaneous (0.5 to 30 cycles), momentary (30 cycles to 2 seconds), temporary (2 seconds to 2 minutes) and



(a) Impulsive transient. **(b)** Oscillatory transient.

Figure 2.11: Example of transients [Seymour, 2001].

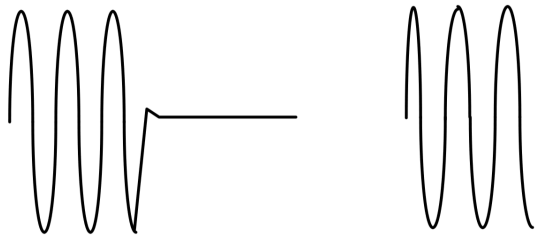


Figure 2.12: Example of a momentary interruption [Seymour, 2001].

(a) Sag. **(b)** Undervoltage.

Figure 2.13: Examples of sag and undervoltage [Seymour, 2001].

sustained (longer than 2 minutes). You might have experienced an interruption at home, causing all lights to go out for some time before coming back. The consequences may be a lot more severe for a manufacturer that is dependant on having a reliable power source.

Sag / Undervoltage. A sag (See Figure 2.13a) is a reduction in voltage that lasts for 0.5 cycles up to a minute. Causes can for instance be the startup of equipment that consumes large amounts of power, or just the system not being able to delivered enough power. Undervoltages (See Figure 2.13b) are the results of sags that have lasted for longer than one minute and can lead to serious damage of equipment. Both sags and undervoltages may be discovered by looking at the RMS value as it will decrease.

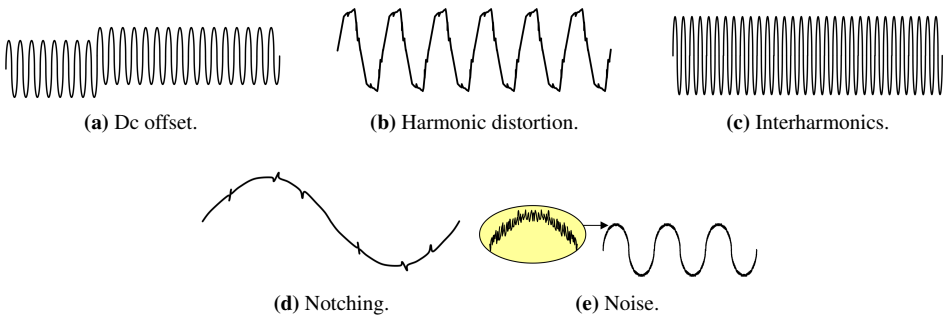
Swell / Overvoltage. A swell (See Figure 2.14a) is the opposite of a sag, that is to say an increase in the voltage that lasts for 0.5 cycles up to a minute. Causes can for instance be the shutdown of equipment that consumes large amounts of power, or faulty isolation. Overvoltages (See Figure 2.14b) are similarly the results of swells that have lasted for longer than one minute. Both swells and overvoltages may be discovered by looking at the RMS value as it will increase.

Waveform distortion is defined as any disturbance that affect the wave of the voltage/current, and can further be divided into five subcategories: DC offset, harmonic distortion, interharmonics, notching and noise.

DC offset (See Figure 2.15a) is an offset that results in the average of the wave not being

(a) Swell. **(b)** Overvoltage.

Figure 2.14: Examples of swell and overvoltage [Seymour, 2001].



(a) Dc offset. **(b)** Harmonic distortion. **(c)** Interharmonics.

(d) Notching. **(e)** Noise.

Figure 2.15: Examples of waveform distortions [Seymour, 2001].

zero, increasing or decreasing the RMS depending on the value of the offset. It is often caused by failure in AC to DC converters, and may result in overheating of the transformers.

Harmonic distortions (See Figure 2.15b) are disturbances in the harmonics excluding the 1st harmonic (the fundamental frequency). Symptoms are for instance overheating in components and loss of synchronization on timing circuits.

Interharmonics (See Figure 2.15c) are a type of distortion that occur when a signal that is not a harmonic is imposed on the wave. Symptoms are for instance overheating in components and flickering lights.

Notching (See Figure 2.15d) is a periodic voltage disturbance. It is similar to the impulsive transient distortion, with the difference being that notching is periodic and as such considered a waveform distortion.

Noise (See Figure 2.15e) is unwanted voltage/current which is superimposed on the wave. Noise may be caused by poorly grounded equipment. This results in the system being more susceptible to interference from nearby devices. Common problems caused by noise are for instance data errors and hard disk failures.

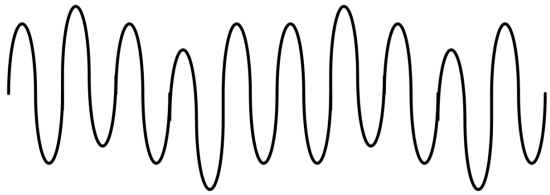


Figure 2.16: Example of a voltage fluctuation [Seymour, 2001].

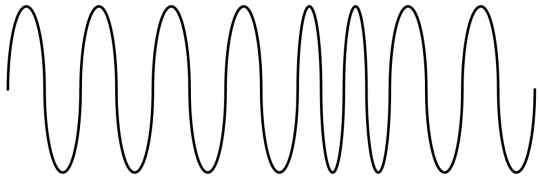


Figure 2.17: Example of a frequency variation [Seymour, 2001].

Voltage fluctuations are series of minor, random changes in the wave of the voltage (See Figure 2.16). The variation is usually between 95% and 105%. The cause is usually a load exhibiting significant current variations. This can for instance result in flickering lights and/or loss of data. A way to resolve this problem is to remove the offending load.

Frequency variations are variations of the frequency in the wave (See Figure 2.17). They are an extremely rare type of waveform distortion. They are usually caused by an overloaded generator and can cause problems like system halts and flickering lights. A way to resolve this problem is to fix the generating power source.

Chapter 3

Background - EarlyWarn

In this chapter we will introduce the EarlyWarn project that this report is a part of. This chapter is mostly based on (sources from) [Santi, 2019].

3.1 Introduction

The main purpose of EarlyWarn is to develop surveillance systems that discover and identify faults and disturbances in the Norwegian power grid, including the distribution-, the regional- and the transmission net. It is crucial that the faults and disturbances are discovered before they evolve into larger problems like power outage, or cause damage to valuable equipment in the power grid and/or equipment belonging to the end consumers. There are many parties involved in this project, including several power grid operators, with the most notable parties being SINTEF[1] Digital and Statnett. SINTEF receives data from various sensors placed all around the power grid from the participating power grid operators. The data is then processed and fed into machine learning and statistical models in order to make predictions and classifications. The desirable outcome is to get a prediction with a **high accuracy**, and in **good time** before the prospective fault. With **high accuracy**, we mean that when a fault is predicted, we are almost completely certain that the fault will occur and that is has to be addressed. With **good time**, we mean that when we get the prediction, we get it sufficiently in advance such that we have time to react, inspect and understand the situation, and then take the necessary measures. The measures could be to reroute the power around the area of the grid that is affected by the fault(s), send a maintenance team to inspect the part of the power grid in question (and to perform

[1] An independent research organization headquartered in Norway that conducts contract research and development projects.

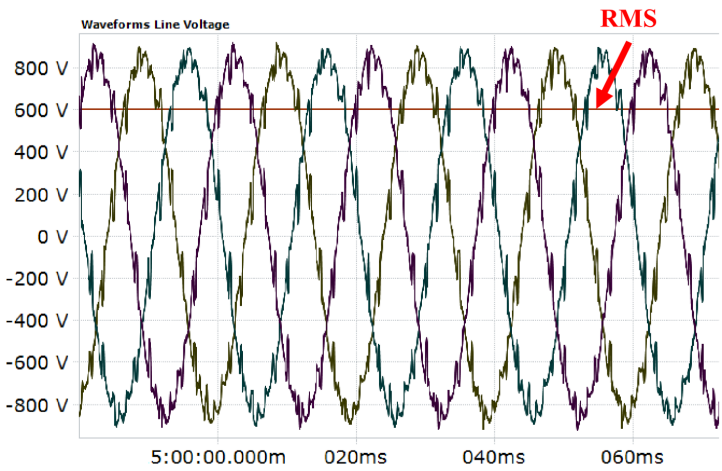


Figure 3.1: Example of a RMS value with its wave affected by harmonic distortion sampled by a PQA [C. A. Andresen and Uhlen, 2018].

repairs if needed), or to simply shut down parts of the power grid in order to prevent the fault(s) from doing damage to the system.

3.2 PQA/PMU sensors

There are mainly two types of sensors utilized in the power grid; Power Quality Analyzers (PQAs) and Phasor Measurement Units (PMUs). The main difference is the frequency of the sampling rate. The PQAs have a sampling rate of up to 25kHz and higher, while PMUs have a sampling rate of just 50Hz [C. A. Andresen and Uhlen, 2018]. This is important to consider as the higher sampling rate makes it possible to detect distortions that would otherwise get lost by using PMUs that have a lower sample rate. Another difference is the data that is collected. PQAs collect data containing information covering all voltage quality parameters, e. g. voltage variations, transients, harmonic distortions as described in Section 2.3. PMUs on the other hand provides phasors, as described in Section 2.2.2, constituted by an angle and a magnitude.

There are multiple pros and cons with both PQAs and PMUs, and they are both useful in different situations. The higher resolution makes the PQAs the preffered option over PMUs in regards to fault detection. By looking at all the different voltage quality parameters, faults and disturbances that are not possible to discover by looking at the RMS value alone can be found (See Figure 3.1). PMUs can be be synchronized very accurately using GPS-signals [C. A. Andresen and Uhlen, 2018] and is therefore very useful for comparing signals at different locations and monitoring at the transmission-net level (See Figure 3.2).

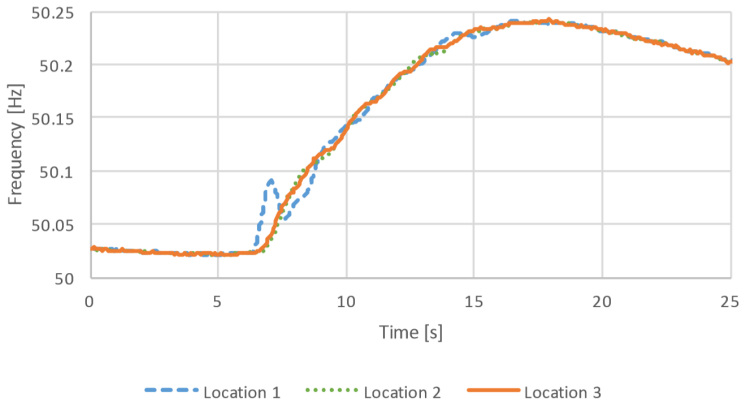


Figure 3.2: Example of frequencies from three locations sampled by a PMU. The frequency suddenly increases at Location 1 because of a loss of load [C. A. Andresen and Uhlen, 2018].

There is also one more important factor that must be considered, there is a downside to the higher sampling rate of the PQAs; the high data sampling rate requires compression/decompression methods when storing/reading, which adds a time-delay. Depending on the application that uses the data, this might be inconvenient. For instance real-time applications are time-sensitive and rely on receiving the data as soon as possible. This is especially true when predicting faults in the power grid. The time window that the operator has to react to might already be very small, thus it is important that it is not made unnecessarily smaller by having to spend time waiting for the data to get processed. PMUs are suited to this as the transfer protocol that is used has a very low latency and the data can be streamed live from the sensors.

As of now all the sensors send the data to a centralised server that stores the time series for all the participating power grid operators. This adds another point of delay as the server has to process the data from all of the sensors. This might be improved in the future as newer sensors [ElspecLTD, 2019] have the capability to process the data themselves before transferring it to the server, saving the server for a lot of processing time.

3.3 Datasets

To extract time series from the centralised server, SINTEF made an application called *Dynamic Dataset Genererator* (DDG). This application lets the user specify a set of parameters in order to extract the desired data (See Table 3.1). The server contains time series for voltages, currents, active- and reactive power, which are all aggregated by a method, for a resolution, both set as two of the parameters. The RMS value, the waveform of the original signal, and up to the 512th harmonic can also be extracted.

Parameter	Description
Total duration	Time duration to include in the sample, before the fault occured.
Resolution	Sampling frequency of the signal in the generated sample.
Buffer	Time duration to include in the sample, after the fault occurred.
Transients	Minimum duration of time that should pass after a fault, before a non-faulty data sample can be generated.
Aggregation method	Method used to aggregate the time series data, when the data extraction sampling frequency is not equal to the original signal sampling frequency. Can choose between *Min*, *Max* and *Average*.

Table 3.1: Parameters for the DDG [Santi, 2019].

To label the extracted time series SINTEF created an analytical tool *A-HA* (automatisk hendelsesanalyse - automatical incident analysis). The tool analyzes time series for a given interval and returns the amount for each of four types of faults; voltage sags, grounding faults, interruptions and rapid voltage changes. A-HA is further able to differentiate between real and false voltage sags. The application creates a list of all the incidents which also contains references to the actual raw data such that deeper analysis may be done if deemed necessary. To balance the dataset with both faults and non-faults, the DDG is also

Parameter	Description
Fault detection	Fault or non-fault.
Fault type	Fault type, if any.
Fault time	Time of occurrence.
Start time	Start time of first sensor reading
End time	End time of first sensor reading
Total duration (sec)	Seconds of data in dataset
Total duration (days)	Days of data in dataset
Resolution (ms)	Time interval between each sample
Time buffer (sec)	Time duration to include in the sample, after the fault occurred.
Time transient (sec)	Minimum duration of time that should pass after a fault, before a non-faulty data sample can be generated.
N points	Number of data points for each parameter.
Node	Name of the node from which the sensor data is accessed.
Nominal voltage	The line voltage of the equipment at the fault location.

Table 3.2: Metadata per sample [Santi, 2019].

able to generate non-fault time series at a ratio given by the user. There is also metadata for each of the samples in the dataset (See Table 3.2).

3.4 False negatives and false positives

Lastly, false negatives and false positives must be addressed. Generally, a false negative occurs when a system predicts that something is false, but in reality it is true. Similarly a false positive occurs when the system predicts that something is true, but in reality it is false. The consequences of both are different depending on the situation and the severity of what being evaluated to true and false. In the context of faults and disturbances in the power grid, a false negative could be when the system predicts that there are no faults and all is good, but suddenly a power interruption happens. A false positive could be when the system predicts that there will be a voltage sag soon, but nothing happens. In the case of the false negative the power grid could get damage that could be prevented if the system was able to predict that the interruption was going to happen. In the false positive case, the power grid operator might have wasted time doing preemptive measures against the voltage sag which was never going to happen. By wasting time on false warnings the operator might also lose confidence in the system, leading to the operator ignoring future warnings. One must have to evaluate the cost of both and compare them. On the extreme side one could avoid all false negatives by always saying there will be a fault, and avoid all false positives by saying there are no faults at all. Saying there are no faults would be the same as not having the predicting system at all. This means that all correctly predicted faults serve as an added bonus, while all wrongly predicted faults serve as added cost compared to the original system. As such, one could argue that reducing the amount of false positives are of higher importance than reducing the amount of false negatives.

Chapter 4

Background - Machine Learning

In this chapter we will briefly explain what machine learning is in general with empathize on the parts that are relevant to our research. We will also show some popular methods.

4.1 Introduction

Machine learning can in short be described as to **learn** from data. A common definition of what it means to learn was defined by T. Mitchell as:

> A computer program is said to **learn** from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E. [Mitchell, 1997]

Over the past decade machine learning has become increasingly popular and has become a popular topic of research. There are various causes for this recent focus. The access to huge amounts of data through the internet to train models on. Many have started to actively collect and process data which have led to a big scope of various data sets, available both publicly and privately. The advancement of computing components has also led to experimenting with increasingly complex models, something which was not possible previously due to the lack of processing power and memory at the time. The transition from doing computations on the CPU to the GPU has also made a huge impact in the processing speed which has allowed the training of wider and deeper models. As GPUs are very good at simple calculations on vectors/tensors they are a perfect fit for training machine learning models. The recent research on machine learning has revealed increasingly efficient

and accurate algorithms and models, meaning that many complex problems now may be solved in real time, putting a spotlight on the field from the commercial sector.

The study of machine learning typically involves developing algorithms and statistical models which learn patterns and intrinsic properties in some data, with the goal of solving a particular problem related to that data. This is in contrast to the traditional way of problem solving, which was to use explicit instructions created by humans. By having the machines discover the features and connections between data points automatically, the process gets added benefits, such as, being less prone to human errors, the possibility of discovering properties difficult/impossible for humans to find, time saving. The downsides include; demands large amounts of data, needs a lot of processing power and memory, needs specialists to create and adjust models.

Machine learning tasks are often split into three main categories.

- **Supervised learning**, where the model is provided with a data set with known categorizations, known as labels of the data. These labels are used to evaluate the predictions, as the model learns by evaluating the difference between the predictions it makes and the label of each data point.
- **Unsupervised learning**, which is characterized by performing machine learning algorithms on data without labels. By finding similarities between data points, hidden structures and patterns may be discovered, despite the lack of explicit feedback of correctness.
- **Reinforcement learning**, which is characterized by the learner being given a reward at various points in its learning process, based on the actions it has chosen. The rewards are given based on a metric independent of the learner, and may both be positive and negative.

4.2 Data and Generalization

Without good and/or enough data it is not possible to sufficiently train the model, let alone give valuable output. Not only is the amount of data important, the model must also generalize well to get a good result. That the model generalize well means that the model will be able to yield good results on new, unseen data, not only on the data it has been trained on. How to process the data and generalize the model will now be discussed further.

4.2.1 Feature Engineering

Feature engineering is the concept of how to process the data into features that the model can learn from. This can either be done manually by domain experts or automatically by

feature learners. There are many different methods involved in feature engineering.

- **Augmentation**, which is a group of methods that lets you increase the diversity without collecting new data. This can be done by generating new data based on the data that is already collected. A vital point is that the newly generated data should have the same label as the data it was generated from, therefore the augmentation method that is used must be label-preserving. Example of such methods are random horizontal flips, cropping, small rotations, illumination changes.
- **Extraction**. There might also be situations where we have a huge data set and only a part of it is relevant to our task. Extraction encompasses methods on how to evaluate what data is relevant as to then retrieve said data.
- **Imputation**, which helps with the handling of missing values. This can be done as easy as to just drop the data which has any missing values, or the missing values can be inferred based on the existing values and/or other data.
- **Transforming**, which transforms the data into a format that makes it easier/possible for the model to learn from. If we for instance have a problem where we want to group a set of data points, but they are not separable in the current representation. We can transform the data into a representation in which they are separable and then group the data. This can for example be done by doing polynomial transform, one-hot encoding, log transform and discrete fourier transform.

4.2.2 Model and Parameters/Hyperparameters

There are both models with and without model parameters, called parametric- and non-parametric models respectively. A parametric model defines a set of parameters of a fixed size that is independent of the amount of data. First you define a function, lets say you want to do line regression and choose a function on the form $ax^2+bx+c=y$. Here we have the parameters a, b and c. Said parameters are then estimated to best match the the data and we get a predicative model that may be used to predict new data. The goal is to find a function that is as close to the underlying true function as possible. Benefits of this approach is that it is fast and simple and doesn't require a lot of data in order to give reasonable output. The downsides are that the model is constrained by the predefined function and that the function rarely matches the underlying function. Much used parametric methods include; neural networks [1], naive Bayes and logistic regression.

Non-parametric functions on the other hand do not make any strong assumptions regarding the form of the underlying function, but rather aim to find a good function form based

[1] Even though neural networks do not make any strong assumptions regards the underlying structure which tends to be a hallmark of the non-parametric models, it is considered a parametric model as it uses a fixed number of parameters to build the model, independent of the data size as defined in [Russell and Norvig, 2016]: "*A learning model that summarizes data with a set of parameters of fixed size (independent of the number of training examples) is called a parametric model.*" However, it is still in a "gray area" and many consider it a non-parametric model.

on the data. For instance clustering methods which might not make any assumptions about the data except that similar data are more likely to be closer to each other (based on some distance metric). Benefits of this approach is that the model is flexible as no strong assumptions about the underlying function is made, and that it therefore can fit various functional forms. Much used non-parametric methods include; clustering, support vector machines and decision trees.

Even if a model is parametric or non-parametric, it will have hyperparameters. Hyperparameters differ from model parameters in the way that they external to the model and cannot be estimated directly from the data. They are set in order to help the process of estimating the model parameters. They are often set based on previous experiences/similar problems, and many models have standard values for them, but they might also be set using heuristics and further tuned. Example of hyperparameters are learning rate, number of hidden layers in a neural network and depth of a decision tree.

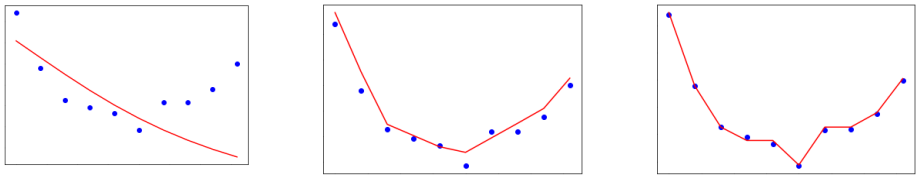
4.2.3 Training, Validation and Testing

The data is usually divided into three parts, a data set for training, a data set for validation and a data set for testing. There is no absolute correct ratio of how to split the data, but it is usual that the training data set is the largest and the test data set is the smallest. The data set is split such that the data that the model learns from are different from the data that it is evaluated on. This is to ensure that the model is not simply just memorizing the data it is trained on, but that it is able to perform well on new unseen data, namely that it generalizes well. The model is first trained on the training data. In this step all the parameters in the model will be fitted as to give the best possible output, thereby requiring the biggest amount of data. In the validation step only the hyperparameters are tuned while the model parameters are frozen. Finally the model is evaluated against the test data set.

4.2.4 Overfitting and Underfitting

One of the most encountered problems in machine learning is overfitting (high variance). This occurs when the model starts to memorize the data it is trained on rather than learning the underlying function. This often results in a complex function with more parameters than needed that is extremely good at representing the training data, but terrible at predicting new unseen data. On the opposite side we have underfitting (high bias). This happens when the model does not have the capacity to learn the underlying function and results in a very simple function that does not have enough parameters and is bad at predicting both training data and new data. A model that is either overfitted or underfitted is a model that generalizes poorly. To get a model that performs well one should get a good balance between variance and bias (See Figure 4.1).

There are many different approaches one can take to reduce overfitting and better generalize ones model.



(a) Underfitted. **(b)** Balanced. **(c)** Overfitted.

Figure 4.1: Example of an underfitted, balanced and overfitted model.

- **Increase amount of data**, the more data the model has to train on, the better chance it has to learn the underlying function. If the data set is small, it is easier affected by noise and might not be representative of the underlying function (See Figure 4.2). This may be done by data augmenting and collecting new data. Data augmenting can only increase the data to a certain extent before the added data gets too redundant and does not add any new value. You could also collect new data, which might be the best way to reduce overfitting, but as it usually is expensive and time consuming it is not always an option.
- **Regularization** adds a penalty to large parameter values in the model. The penalty is added as an addition to the loss function of the model. As large parameter values are punished, complex models are discouraged which reduces the risk of overfitting. The parameter values tend toward zero which results in a more sparse model that more easily learns the relevant patterns in the data. The most used regularization methods are L1- and L2 regularization, which add the absolute- and the squared value of the parameter as the penalty respectively.
- **Early-stopping**. A common learning approach is an iterative learning process. In an iterative approach we have a repeating process where we at each step take a small step in the direction that minimizes the loss. By doing this we minimize the error in the final model as the iteration enables the model to correct itself whenever there is

(a) Insufficient data. The data set is not representative of the underlying function.

(b) Sufficient data. The data set is able to capture the underlying function.

Figure 4.2: Example of insufficient and sufficient data.

an error. However, there is a point in this repeating process where the model stops to learn new information about the underlying structure from the training data, and rather starts to memorize it (assuming the model has sufficient capacity). The model should stop training when it reaches that point, but it is not always trivial to know when to stop. One option is to monitor the gap between the accuracy of the training and validation data and to continue to train as long as the gap decreases, but stop when the gap is not changing or starts to increase. The gap is also referred to as the generalization error (which is a measure of how good a model is at predicting new unseen data, i.e. how well the model generalizes).

4.2.5 Feature Normalization

Feature normalization is the process of scaling data from the original distribution and region to a predefined distribution and/or region. One common way is to remove the mean from the data and scale the variances to unit distance, creating an approximately normally distributed data set. Another is linear scaling, where the data preserves the distribution of the original data, but is scaled to unit range. Feature normalization can improve the results of algorithms which base themselves on distance between data points, as it gives the same region of possible values for all features.

4.2.6 Dimensionality Reduction

Dimensionality reduction is a process which creates new features that best preserve the information stored in the original features of the data set. It does this by finding a set of principal variables, which represents the features in the original data as close as possible. The number of new features does not surpass the number of original features, and there are usually many fewer new features than there were originally. How closeness between the original and reduced data is calculated depends on the method used.

Principal Component Analysis

Principal Component Analysis, hence called PCA, is the simplest form of linear dimensionality reduction. [Bengio et al., 2013] defines PCA as "[Something that] learns a linear transformation $h = f(x) = W^T x + b$ of input $x \in \mathbb{R}^{d_x}$, where the columns of $d_x \times d_h$ matrix W form an orthogonal basis for the d_h orthogonal directions of greatest variance in the training data. The result is d_h features (the components of representation h) that are decorrelated." In other words, it linearly transforms a d_x dimensional data set into a d_h dimensional one, by the use of an orthogonal matrix W. Due to learning a linear transformation, PCA has a rather lacking expressive power, and fails to find deeper correlation in data. PCA uses *explained variance*, to estimate how much information is retained in the

reduced data. The variance – $\mathrm{Var}(X)$ – of all feature columns are calculated, and normalized – $\mathrm{Var}(X_i) = \frac{\mathrm{Var}(X_i)}{\sum_j \mathrm{Var}(X_j)}$ –. The columns with the highest explaining variance are combined until the output has a summed greater variance than the variance which needs to be explained explained.

t-distributed Stochastic Neighbor Embedding

t-distributed Stochastic Neighbor Embedding, hence called t-SNE, is a non-linear dimensionality reduction technique. The output is commonly two or three dimensions, and as such is well suited for visualization of high-dimensional data. t-SNE gives each data point a location in a lower-dimensional map where each data point is positioned so that it is similar to data points close to it, and dissimilar to data points far away, with a high probability. The goal of t-SNE is to minimize the Kullback-Leibler divergence $C = \sum_i KL(P_i||Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}}$ where $p_{j|i} = \frac{\exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^2/2\sigma_i^2)}{\sum_{k \neq i} \exp(-\|\mathbf{x}_i - \mathbf{x}_k\|^2/2\sigma_i^2)}$ and $q_{j|i} = \frac{\exp(-\|\mathbf{y}_i - \mathbf{y}_j\|^2)}{\sum_{k \neq i} \exp(-\|\mathbf{y}_i - \mathbf{y}_k\|^2)}$. Here $x = \{x_1, x_2, ..., x_n\}$ is the original n data points in the high dimensional space, and $y = \{y_1, y_2, ..., y_n\}$ is the resulting n data points in the low dimensional space.

It does this by first constructing a probability distribution $p_{j|i}$. In other words, the probability of data point j being chosen as a neighbour of point i, proportional to the distance of all other points in the data set. σ_i is the Gaussian variance, which must be set to some reasonable value by the user upon initialization, depending on the sparsity and distance in the data set. This makes it so that similar points are likely to be chosen as neighbors, while dissimilar will have an almost infinitesimal chance of being chosen as neighbors. This is calculated for all pairs of the high-dimensional data. These probabilities are used to decide whether two points are neighbours or not. It then constructs a low dimensional space with probability distribution $q_{j|i}$ with similar amount of points as in the original data. It then minimizes the Kullback-Leibler divergence using gradient descent, i.e. it changes Q as to minimize C. Once the Kullback-Leibler divergence has been minimized, Q is considered be the probability distribution that loses the least information entropy when representing P, i.e. it best represents the neighborhoods present in the original data, in the chosen dimensions [van der Maaten and Hinton, 2008].

4.3 Machine Learning Methods

4.3.1 Clustering

Clustering is a form for unsupervised learning which is used to find underlying patterns, structures, and intrinsic properties in some data by categorizing similar objects into groups, called clusters. Clustering is not one singular algorithm, but rather a general task to be pre-

formed. As such there are many algorithms which might satisfy the task, commonly shared amongst these is the notion of small distances between cluster members, and assumptions of the distribution of data, most often an assumption of cluster denseness or a kind of statistical distribution in each cluster.

The distance – also called dissimilarity – measure between two points in the data space is denoted as $D(x, y)$, which has the following properties.

- $D(x, y) \geq 0 \ \wedge \ D(x, y) = 0 \Leftrightarrow x = y$
- $D(x, y) = D(y, x)$
- $D(x, y) \leq D(x, z) + D(z, y)$

The most common distance measure is the squared euclidean distance, $D(x, y) = \sum_{j=1}^{P}(x_j - y_j)^2$ in a P-dimensional space where $x = (x_1, ..., x_P)$ and $y = (y_1, ..., y_P)$, and will be the distance measure used in the rest of this report unless otherwise stated.

As discussed in [Aggarwal et al., 2001] and [Domingos, 2012], euclidean distance fails to separate data of higher dimensions, and as such the clustering methods using such distance measures will require the data to first be dimensionally reduced, as presented in Section 4.2.6, to yield significant results. The data will also need to be normalized to a shared value space, as presented in Section 4.2.5, so no single feature has a bigger impact on the euclidean sum than other features.

k-means Clustering

k-means is a clustering method which partitions *n* data points into *k* clusters, where each data point belongs to the cluster which has its centroids closest to the data point. This splits the data space into a Voronoi diagram with the centroids as the basis for the Voronoi cells, as demonstrated in Figure 4.3. A Voronoi diagram is a diagram where a set of points is given, and each subspace in the diagram is seen as to belong to the point closest to it, the subspace owned by one point is called a Voroni cell. A centroid is defined as the mean of all the data points in a cluster. The Voroni cells are considered the clusters found in the data. As a consequence of finding the Voroni cells for each centroid, the *k*-means algorithm find clusters which are normally distributed around the centroids, and assumes a circular (or hyper-spherical for higher dimensional data) shape of the clusters. Since single points changing clusters have more influence on positions of clusters containing few points *k*-means clustering also assumes that the clusters contains an approximately similar amount of data points [Miyamoto et al., 2008].

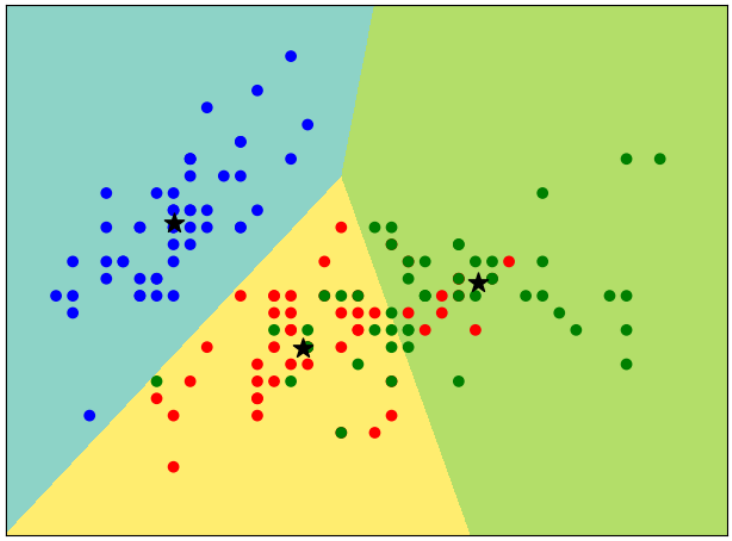


Figure 4.3: *k*-means clustering on the first two dimensions of the iris data set. The background displays the found clusters, while the points are coloured based on their actual labels. Centroids are marked with black stars.

4.3.2 Gaussian Mixture

Mixture models are probabilistic model which incorporates multiple statistical models to predict the mixture model's final output. Mixture models are well suited at representing data sets when the models they are made out of can be used to represent smaller parts of the data set, despite the complete data set not being presentable for any statistical distribution. A Gaussian mixture model is a mixture of Gaussian distributions, and is well suited for finding multiple Gaussian distributions in a data set by $p(\mathbf{X}) = \prod_{n=1}^{N} p(\mathbf{x}_n) = \prod_{n=1}^{N} \sum_{k=1}^{K} \pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k)$. The formula presents the relation between the data set $\mathbf{X}$ and the amount of Gaussians in the mixture K. π_k, μ_k and Σ_k are the weight, mean, and covariance of the k^{th} cluster, respectively. Gaussian mixture models are ideal for clustering of Gaussian data, including non-spherical data, where *k*-means fails. Since Gaussian mixture is a combination of probability distributions, each point is priorly given an estimated probability to belong to each cluster, $\pi_k \mathcal{N}(\mathbf{x}_n|\mu_k, \Sigma_k)$, and as such the Gaussian mixture model performs what is called fuzzy clustering. Fuzzy clustering is where each data point has a partial membership to each cluster, with a higher membership to clusters which better represent it. This stands in contrast to crisp clustering, such as *k*-means clustering, where each data point is defined as to belong only to one cluster. By being fuzzy, the Gaussian mixture model is capable of expressing what certainties it thinks each point is correctly clustered, as a data point without any strong belonging to any cluster might be indications of a split cluster, noise or poorly separable clusters.

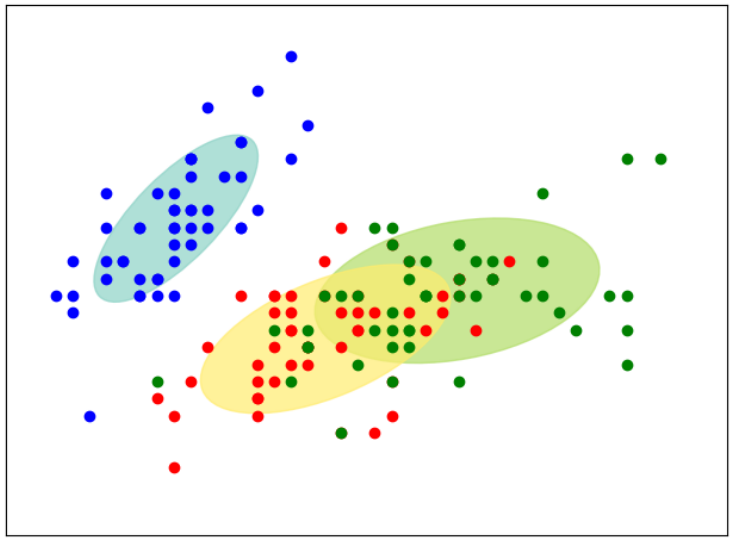


Figure 4.4: Gaussian mixture on the first two dimensions of the iris data set. The ellipses displays the Gaussian distributions found, while the points are coloured based on their actual labels.

The Gaussian mixture model is usually optimized by the use of the expectation-maximization algorithm. The expectation-maximization algorithm, hence called EM, is an iterative method which finds the maximum likelihood estimates of parameters in a statistical model, where the model depends on latent variables. The maximum likelihood estimates of parameters in a model to some observed data means that the parameters of the model are tuned as to most correctly produce that observed data, by maximizing some likelihood function. Latent variables are variables which are not directly observed, but rather inferred from other observable variables. The EM algorithm consists of two steps: The Expectation (E) step, where the likelihood between the data and the current parameters are calculated, and the Maximization (M) step, where the likelihood found in the E step are maximized, and new maximum likelihood estimates of the parameters are found. These estimates are then used in the next E step, until convergence. By using the EM algorithm on the Gaussian mixture model, the means and variances of all the Gaussian distributions in the Gaussian mixture model are tuned as to best predict the observed data given, making it good at clustering Gaussian data.

4.4 Evaluation Metrics

There are many way to evaluate performance of machine learning methods. Here we present the ones used in this report.

4.4.1 Matthews Correlation Coefficient

Matthews Correlation Coefficient, hence called MCC, can be used to measure the quality of binary and multiclass classifications. It takes into account the true positive, the false positive, the true negative, and the false negative amounts, referred to as TP, FP, TN, and FN respectively henceforth. It is regarded as a balanced measure which can be used even if the classes are of very different sizes.

MCC is a correlation coefficient value in the interval [-1, 1] where 1 indicates a perfect prediction of the data, -1 a perfect inverse prediction, and 0 is an average random prediction. MCC is calculated as such:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

If any of the four sums in the denominator is zero, then by definition $MCC = 0$ [Powers and Ailab, 2011].

4.4.2 Receiver Operating Characteristic Curves

Receiver Operating Characteristic metrics, hence called ROC, can be used to measure classifier output quality. The curve is created by plotting the percentage of TP per percentage of FP found by the classifier, i.e. TPR (True Positive Rate) on the Y axis and FPR (False Positive Rate) on the X axis of a 2d plot. This means that being in the top left corner is the ideal point, where FPR is 0 while TPR is 1. As this is not a realistic goal for any classifier, the Area under the Curve (AUC) is calculated to evaluate the classifier. A greater AUC is associated with a better classification. A perfect prediction (TPR=1 when FPR=0) is used as a perfect score with an AUC of 1, and a linear scaling (FPR = TPR) is used as the baseline as a truly random classifier, with an AUC of 0.5 [Zweig and Campbell, 1993].

4.4.3 Silhouette Score

Silhouette is a method of interpretation and validation of consistency of clustering of data. Silhouette analysis can be used to study the separation distance between the resulting clusters, where the silhouette value is a measure of how similar an object is to its own cluster compared to other clusters. Silhouette scores are values in the interval [-1, 1], where a high positive value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters, a low positive value indicates uncertain matching to its cluster, and a negative value indicates it being in a poorly chosen cluster. If most objects have a high value the clustering configuration can be considered appropriate. If many points have a low or negative values there might be too few or too many clusters.

The silhouette is defined by the following equation:

$$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$$

where *s(i)* is the silhoutte score of point *i* and *a(i)* and *b(i)* is the mean distance between *i* and all other points in the same cluster, and between *i* and all other points in the closest other cluster respectively, given by the following equations:

$$a(i) = \frac{1}{|C_i| - 1} \sum_{j \in C_i, i \neq j} D(i, j)$$

$$b(i) = \min_{k \neq i} \frac{1}{|C_k|} \sum_{j \in C_k} D(i, j)$$

where C_i is a cluster and $D(i, j)$ is the distance as defined in Section 4.3.1. Due to this it is better at finding silhouettes for methods using the same distance measure, and might give poor results for clustering methods using another similarity measure, even when the method has found a visually perfect fit for the data [Rousseeuw, 1987].

4.5 Machine Learning on the Power Grid

In our research we will mainly look at unsupervised learning methods, which as previously discussed is useful for finding the underlying structure and intrinsic properties in the data. As the problem we face is related to faults in the power grid, one of our goals is to find the most differentiating features of the faults in order to group similar faults close to each other, and dissimilar faults far away. These features might be voltage, harmonics, or current measured by the sensors. A desired outcome would then be that these features are compared in such a way that faults caused by humans, faults caused by nature, and faults caused by the equipment are clearly separated into their own groups. By discovering these features' importance regarding the faults, we can not only use them to predict faults that might occur in the future, but also take preemptive steps such that these faults are less likely to occur in the first place.

The original data is usually sampled at very high rate which would take too long time to process, not to mention unrealistic to handle real-time. Feature engineering must therefore be done in order to reduce the size of the data. It is important that the underlying structure is preserved as most as possible during this process. By using the Fourier-transform we extract the harmonics and map the signal from the time dimension into the comparably smaller frequency dimension. The data can further be reduced by selecting the most important harmonics while disregarding the harmonics that have a small impact on the clustering results. These selected harmonics may also be aggregated in order to reduce the size even again if necessary.

Chapter 5

Related Work

In this chapter we will introduce and compare related work that has affected the work done in this report. There have not been many attempts at clustering in the context of data related to power grids. Most of the previous works this report is based on was done in collaboration with SINTEF's EarlyWarn project. As we lack resources on clustering of power grid related data, we have looked at work on classification and prediction of faults in the power grid as well.

5.1 Work related to EarlyWarn

Most of the work related to the EarlyWarn project has been about prediction and classification of faults in the power grid. There have been written two master's theses in collaboration with the EarlyWarn project [Santi, 2019] [Høiem, 2019]. Both of them had the shared goal of figuring out if it was possible to predict and classify faults in the power grid using machine learning, and if so, to what extent. Even though their goal differed from the goal in our report in that they research prediction, whereas we researched clustering, we still shared the same data and environment. The background related to power grids in this report has mostly been based on these two theses in addition to the sources mentioned in them. We have also done most of our experiments and exploration on the datasets generated in [Santi, 2019]. We also used the same aggregation methods resulting in the same features in order to remove the time dimension.

There has been some work regarding clustering, most notably a summer project that tried to differentiate fault and non-faults by using t-SNE and k-means clustering. His method was similar to the method used in this report, the difference being in that we tried a wider scope of parameters and a larger dataset both in regards to size and types of faults. In

addition we have experimented with other methods such as GMMs, and evaluated against a broader set of performance metrics. The clustering projects have mostly been for internal use and experiments, and as such there have not been made any publications as of now.

5.2 Data Streams Clustering

Recently big data, and particularly data streams, have gathered more focus due to advancements in technology and the ability to gather more data than before. Instead of manually going around reporting in data, smart-meters and sensors that can automatically report large amounts of data continuously have seen much more use than before. As multiple PQA sensors are placed at many different locations on the power grid and that samples data at a very high rate, we receive a lot of data at a high velocity. It is not possible to fit the entire data stream into main memory, so it has to be stored in secondary storage. Only portions of the stream can be used at a time, which is chosen with a predefined window size. In order to adjust the size of the time window, especially when increasing it, we have to reduce the amount of features by for instance aggregating or choosing a smaller portion. Without doing this we would not be able to fit the data in main memory. Even though the server continuously receives data from the sensors, we mostly cluster the data in an ”offline” fashion. We send queries to the server and are mostly unaffected of the implications that comes with an online stream. The following data stream clustering algorithms we now will look at are mainly useful in the case we choose to go ”online”. However, some discoveries are still interesting and will be considered for future works 7.

In [Mansalis et al., 2018] they evaluated data stream clustering algorithms. One of the assumptions they make is that the objects of the data streams arrive independently. This does not apply to us as the signal we look which is the source of the stream is a periodic waveform and as such is dependant on time. Also, they assume that one is only able to pass over data once. As we in this report do not perform online data stream clustering we are able to pass over the same data multiple times. This would however be important to consider if we wanted to cluster the stream in real-time. In [S. Guha and O'Callaghan, 2000] they presented a partitioning-based algorithm called *Stream*. Partitioning-based algorithms divide the data into a defined set of partitions before clustering over them. *Stream* is based on *k*-medians clustering which is the same as *k*-means except that is uses the medians instead of the means in order to calculate the centroids. The stream is broken up into *n* batches and *k*-medians is applied to each batch resulting in $k \cdot n$ medians. When the number of medians exceed a specific threshold, *k*-medians is again applied over all stored medians to generate a new set of medians. One of the restraints by using this method is that there is no data ageing, i.e. old data is as important as new data. As we currently aggregate the data over a certain time frame this restraint also applies to us and might be interesting to look into.

In [C. C. Aggarwal and Yu, 2003] they presented another partition-based algorithm called *CluStream*. CluStream is a two-phase algorithm. In the first phase it makes summaries

ATTRIBUTE	TYPE	DESCRIPTION
HOUR	Factor	Hour of fault occurred
WEEKDAY	Factor	Weekday of fault occurred
MONTH	Factor	Month of fault occurred
CAUSE	Factor	Direct Cause
EQUIPMENT	Factor	Equipment Involved
COMPONENTS	Factor	Component Involved
CUSTOMERS	Factor	No. of Customers Interrupted
MINUTES_LOS	Factor	Customer Minutes Lost

Table 5.1: Attributes of data used in [Silva and Saraee, 2019].

over the stream. Summaries are different aggregations (like the sum of the data objects) over the data that makes it easy to calculate clustering features (such as centroids by dividing the sum by the number of data objects). In the second phase k-means is used over the summaries in a specified time interval. In [Mansalis et al., 2018] they evaluated CluStream over 4 datasets from [Lichman, 2013] and [Blackard and Dean, 1999]. The evaluation and performance metrics can be found in [Mansalis et al., 2018]. They discovered that the number of summaries should be larger than the number of total clusters, and that the performance increases as the number of summaries increases. Some restraint were for instance that just as *Streams*, old data is as important as new data, and that it did not perform well on high dimensional data and outliers. This makes it less attractive for us as we work with very high dimensional data, and according to work done at EarlyWarn, outliers might be a good implication of faults.

An algorithm that can handle both high dimensional data and outliers was presented in [Lin and Lin, 2009], a density-based algorithm called *HDDStream*. A density-based algorithm uses summaries as partition-based algorithms, but also introduces density properties as radius and center calculated by a weight, which again is given by a time formula that decreases exponentially over time. Because of these density based summaries outliers can be identified as points in regions with very low density. By using a time formula, summaries which contains older data get less weight than summaries with newer data. They discovered that they achieved the best results by having a relatively low weight for older data, and that the performance decreased as the weight increased.

5.3 Power Grid Clustering

As mentioned there have been few attempts at clustering faults and disturbances in the power grid. There have been some, but they mostly use a different type of data than in this report and with other goals. We have mostly used aggregated and transformed data from the original wave signal from some time before the fault occurs. In contrast most other attempts have been using meta data from after the fault has occurred. For instance the data

used in [Silva and Saraee, 2019] can be seen in Table 5.1. The experiment performed on that data also differed in that they before clustering used various text analysis methods on the data resulting in a matrix of text terms. This had to be done as they primarily worked with text values in contrast to the pure numerical values used in our work. The goal if their work was to find out how many people got affected when a fault occurred and to what extent. Other works related to clustering data from the power grid have for instance had goals of trying to find the optimal location of sensors [Kassem and Eissa, 2018], or how generators affect each other [Davarikia et al., 2018], which unfortunately did not use any methods useful for the scope of this report.

Chapter 6

Exploration of Data

This chapter presents the methods used, and results found, during this report. We aim to enlighten the research questions presented in Section 1.2, and reveal potential further areas of research, presented in Chapter 7.

6.1 Data sets

This report will build upon the findings presented in [Santi, 2019], and use the same data sets. These data sets are reintroduced here in Table 6.1, Table 6.2 and Table 6.3 for ease of access. When each data set of faults was generated, a data set of non-faults of the same size was also generated. However, due to some missing time series data, not all fault event time series are available in the database, as such are the number of successfully extracted fault samples included in the data set tables. The tables explain what harmonics are preserved, under Specificity, and how the harmonics were aggregated. For a complete explanation of the creation of the data sets, please see the original thesis [Santi, 2019].

6.2 Features

The features used in this exploration are the same as the ones used in [Santi, 2019]. These are, the mean, the standard deviation, and the maximum for each harmonic, where the harmonics are calculated as explained in Section 2.2.6. The values are calculated over the entire duration, apart from the last 5 minutes, as to find if there exists structural differences between faults and non-faults 5 minutes prior to a fault.

Extraction parameter	**Value**
Total duration	3600
Resolution	1 second
Aggregation method	Mean
Extracted feature	**Specificity**
Single phase harmonics	Phase 1, 2, 3. 1st to 16th harmonic
Line harmonics	Line 12, 23, 31. 1st to 16th harmonic
Fault types	**Successfully extracted**
Voltage sags	2178
Ground faults	1730
Power interruptions	220
Rapid voltage changes	132

Table 6.1: Data set 1.

Extraction parameter	**Value**
Total duration	900
Resolution	1 second
Aggregation method	Mean
Extracted feature	**Specificity**
Single phase harmonics	Phase 1, 2, 3. 1st to 256th harmonic
Fault types	**Successfully extracted**
Voltage sags	2236
Ground faults	1749
Power interruptions	229
Rapid voltage changes	133

Table 6.2: Data set 2.

Extraction parameter	**Value**
Total duration	3600
Resolution	1 second
Aggregation method	Minimum, Maximum
Single phase harmonics	Phase 1, 2, 3. 1st to 16th harmonic
Line harmonics	Line 12, 23, 31. 1st to 16th harmonic
Fault types	**Successfully extracted**
Voltage sags	2178
Ground faults	1710
Power interruptions	220
Rapid voltage changes	132

Table 6.3: Data set 3.

6.3 Methods

The clustering methods used in this report are the ones presented in Section 4.3.1, namely *k*-means clustering and Gaussian mixture models. Different feature sets were created, one where the features kept their original region, and one where the features were scaled to unit distance, to test the effect it would have on the euclidean distance measurements. PCA with 50%, 60%, 70%, 80%, 90%, 95% and 99% explaining variance were calculated, as well as t-SNE with 2 and 3 dimensions, for both feature sets. Between 2 and 20 clusters were tested on each type. All harmonics were used for each data set. For data set 3 an additional test was done, were only the harmonics described in [Santi, 2019] as the most important were used, this data set will henceforth be referred to as *Data set 3+*.

For evaluation of the clustering algorithms ROC-AUC was used as a guiding measure for quality of the clustering. Since ROC-AUC is meant for supervised learning, each cluster was given a predictive value of equal the percentage of faults in the cluster. I.e. a cluster with 3 faults and 9 non-faults classifies a data point which would land in that cluster as 25% probability of being a fault. For Gaussian mixture model the probability of being fault is scaled with the membership for each cluster the data point belongs to. This method is used to filter out clustering methods which have high silhouette scores due to clustering noise vs proper data, where one cluster is significantly bigger than the other, and lacks any predictive use.

To evaluate the existence of fault-only – or non-fault-only – clusters, the silhouette scores for all data points were plotted, grouped by clusters. Each score was coloured to represent whether it was a fault or non-fault.

6.4 Results

The silhouette scores from the best predictive clustering according to their ROC-AUC scores are displayed in Figure 6.4, 6.6, 6.9 and 6.11 for each respective data set. The Matthew correlation coefficients were found to be very small due to the high error-rate of the classifications, and as so were they not used to find the best clusters, as it did not manage to show any significant difference between what AUC-ROC considered better and worse classifiers. The Matthew correlations coefficients and AUC-ROC scores are given in Figure 6.1.

Figure 6.4 displays some major clusters with a 50/50 split between faults and non-faults, and some minor, mostly homogeneous, clusters. These minor clusters contain almost solely faults, which indicates that there are some faults which are considerably separated from the other data points. The silhouette scores are mostly low, with clusters such as cluster 9 being mostly negative, which indicates that the clusters are not well separated, as can be seen in Figure 6.3. This indicates in total that there are some faults which appear separable, but that most of the data is not separable from each other in any meaningful

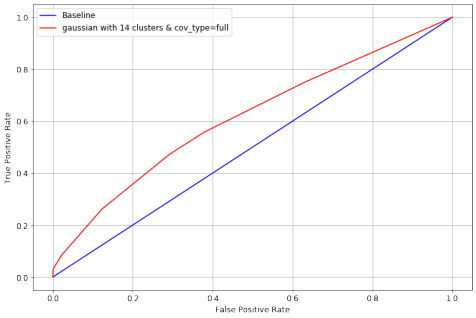
way. That the data is close to inseparable comes clear upon inspection of the data spread, shown in Figure 6.2.

Figure 6.6 shows fairly equally sized clusters, with mostly positive silhouette scores. However, as the ROC-AUC score in Figure 6.1(b) suggests, does it not differentiate well between faults and non-faults. This can also be seen from the silhouette scores, as all the clusters are mostly balanced, with no clusters showing a overwhelming majority of neither faults nor non-faults, something to be expected considering Figure 6.5.

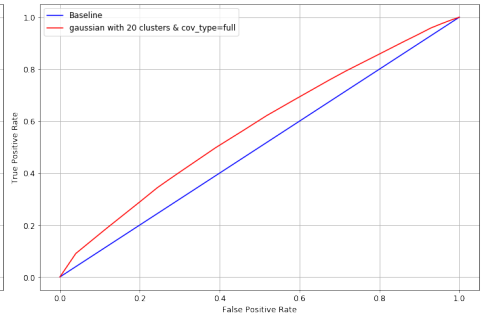
Figure 6.9 is akin to 6.4 in the sense that it has some major clusters, and some minor. The data spread and, as a result, the clustering shown in Figure 6.7 and 6.8 is also similar to that of data set 1. The minor clusters are again mostly faults, while the major clusters are balanced. The similarity suggests that data set 3, similar to data set 1, contains some dimensionally separable faults, but mostly has an inseparable gathering of faults and non-faults.

Data set 3+ has the highest ROC-AUC score, as seen in Figure 6.1(d), suggesting it acting as a good classifier. Figure 6.11 supports this, with two clusters, namely 2 and 11, being majorly faults, in addition to the clusters being fairly well separable, having good silhouette scores. This indicates that there are some structural uniqueness amongst the faults in cluster 2 and 11. This is also apparent by inspecting Figure 6.10. However, most of the data points are still located in balanced clusters.

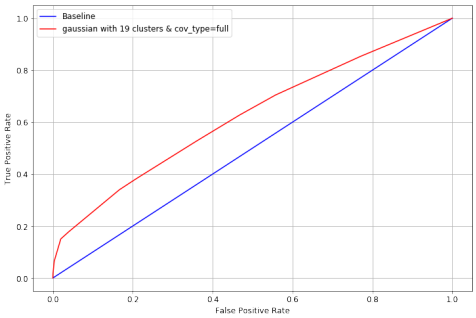
In summary, for the best classifying clusters according to ROC-AUC, only data set 3+ was able to find any structural differences between faults and non-faults, and only for some portion of the faults. This suggests that while some faults are structurally different from other data when using the right features, there are no major intrinsic structural differences between fault and non-faults.



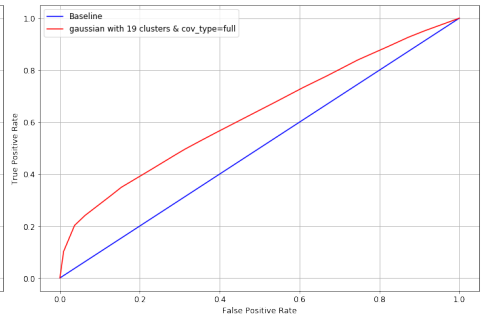
(a) ROC-AUC-score = 0.6224. MCC = 0.0265.



(b) ROC-AUC-score = 0.5813. MCC = -0.0073.

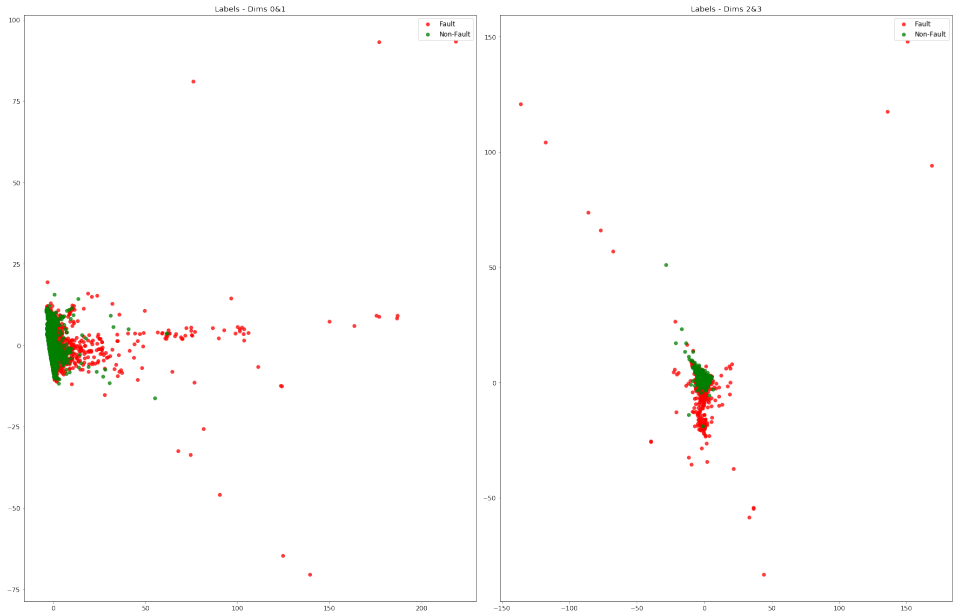


(c) ROC-AUC-score = 0.6280. MCC = 0.0329.

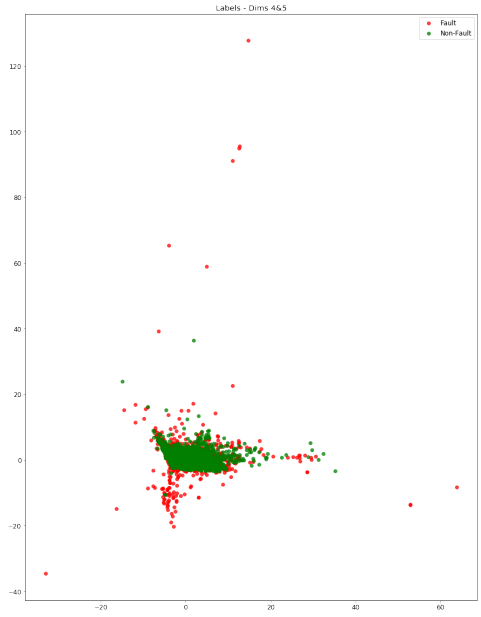


(d) ROC-AUC-score = 0.6334. MCC = 0.0042.

Figure 6.1: ROC-AUC score and curve, and Matthew correlation coefficient of Gaussian mixture. (a) shows for 14 clusters on normalized and PCA reduced data with 60% explaining variance on data set 1. (b) shows for 20 clusters on normalized and t-SNE reduced data with 3 dimensions on Data set 2. (c) shows for 19 clusters on normalized and PCA reduced data with 50% explaining variance on data set 3. (d) shows for 19 clusters on t-SNE reduced data with 2 dimensions on data set 3+.

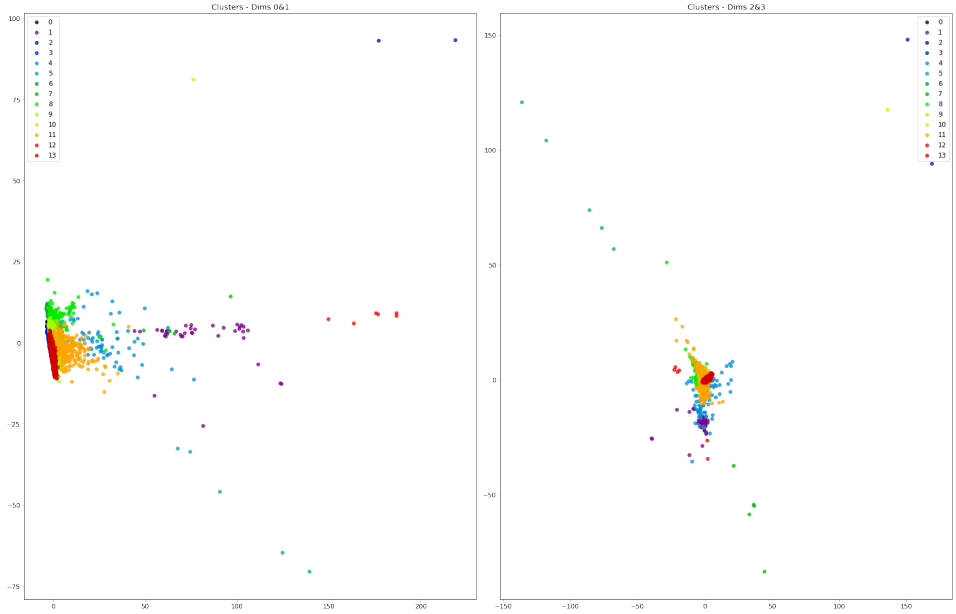


(a) **(b)**

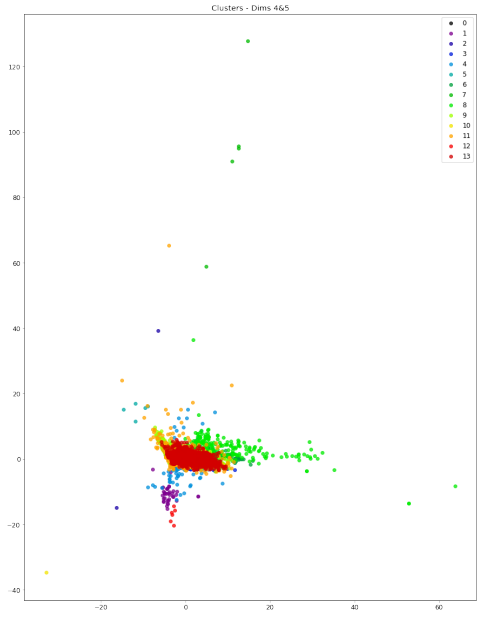


(c)

Figure 6.2: Data spread of normalized and PCA reduced data with 60% explaining variance on data set 1.



(a) **(b)**



(c)

Figure 6.3: Clusters of Gaussian mixture with 14 clusters on normalized and PCA reduced data with 60% explaining variance on data set 1.

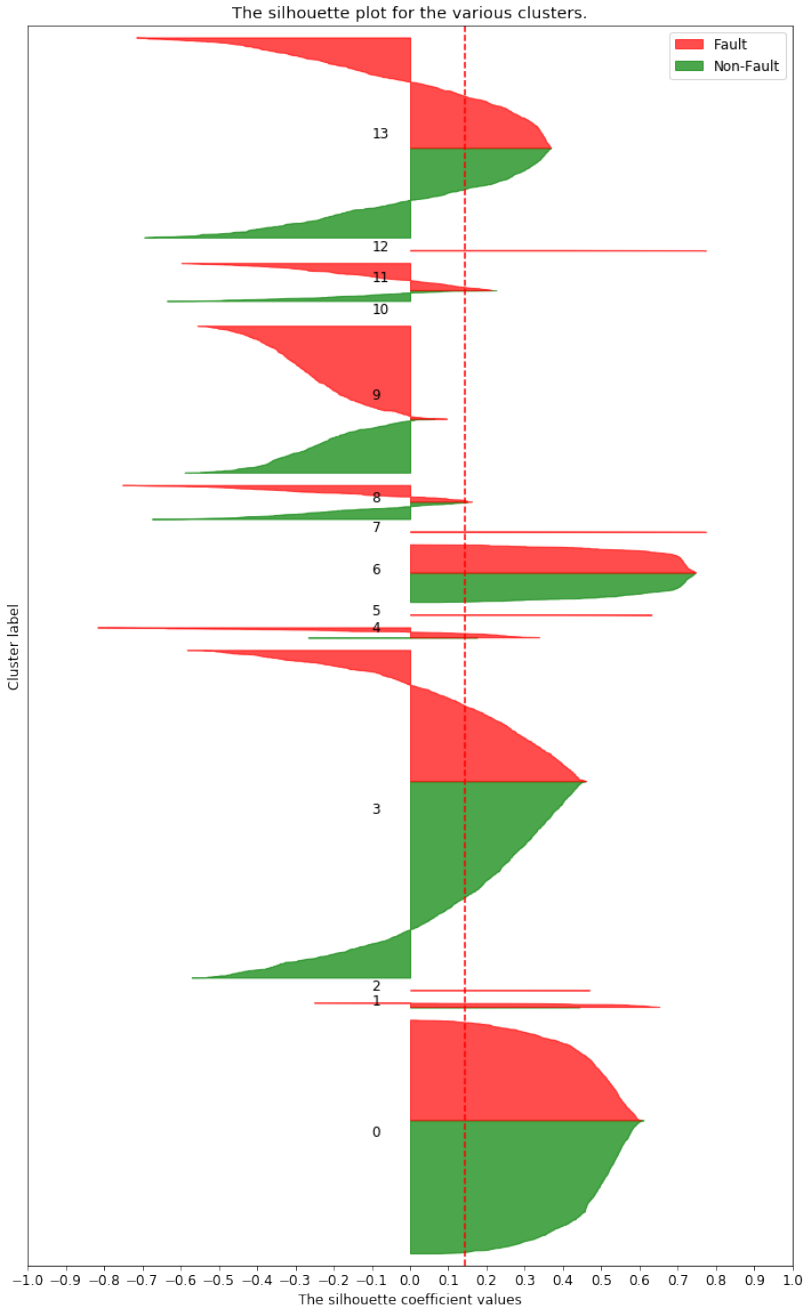


Figure 6.4: Silhouette scores of Gaussian mixture with 14 clusters on normalized and PCA reduced data with 60% explaining variance on data set 1.

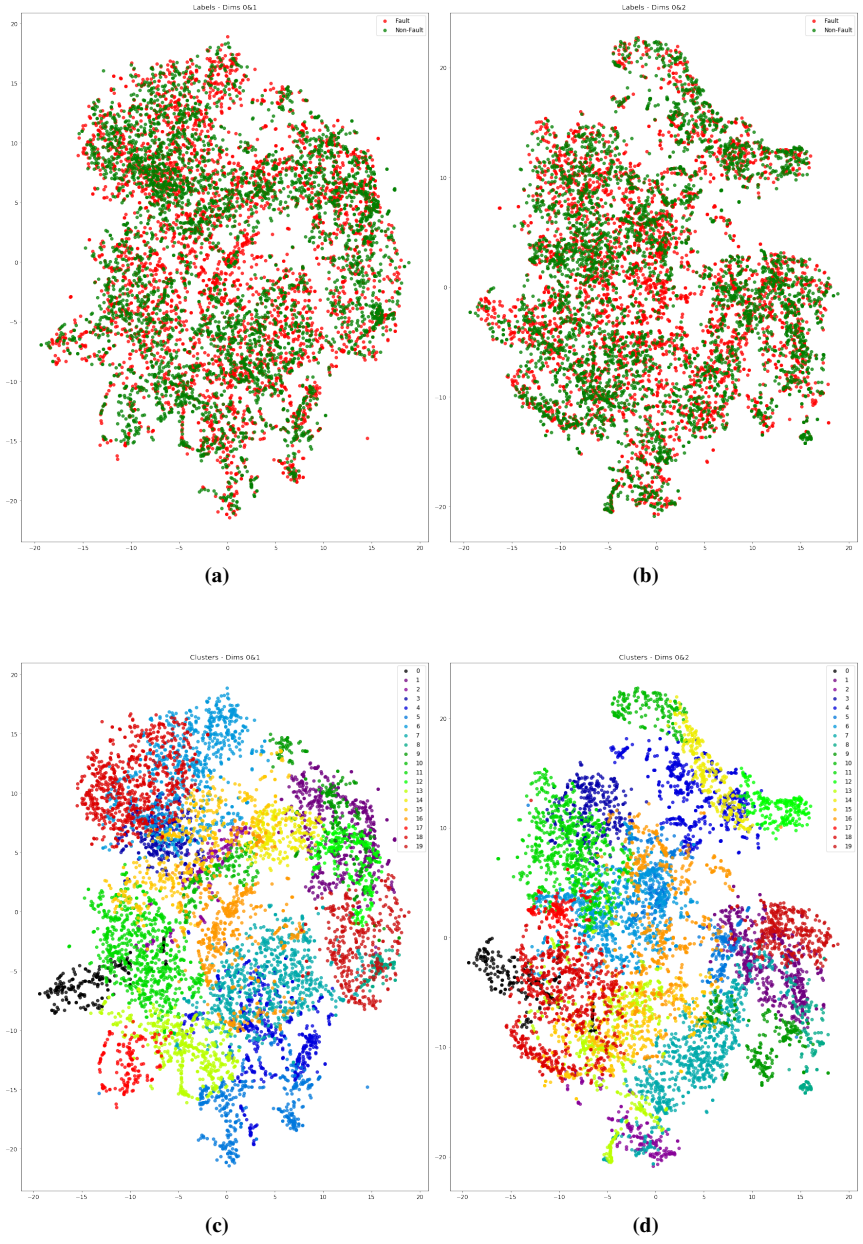


Figure 6.5: (a) and (b) show data spread of normalized and t-SNE reduced data with 3 dimensions on Data set 2. (c) and (d) shows clusters of Gaussian mixture with 20 clusters on normalized and t-SNE reduced data with 3 dimensions on Data set 2.

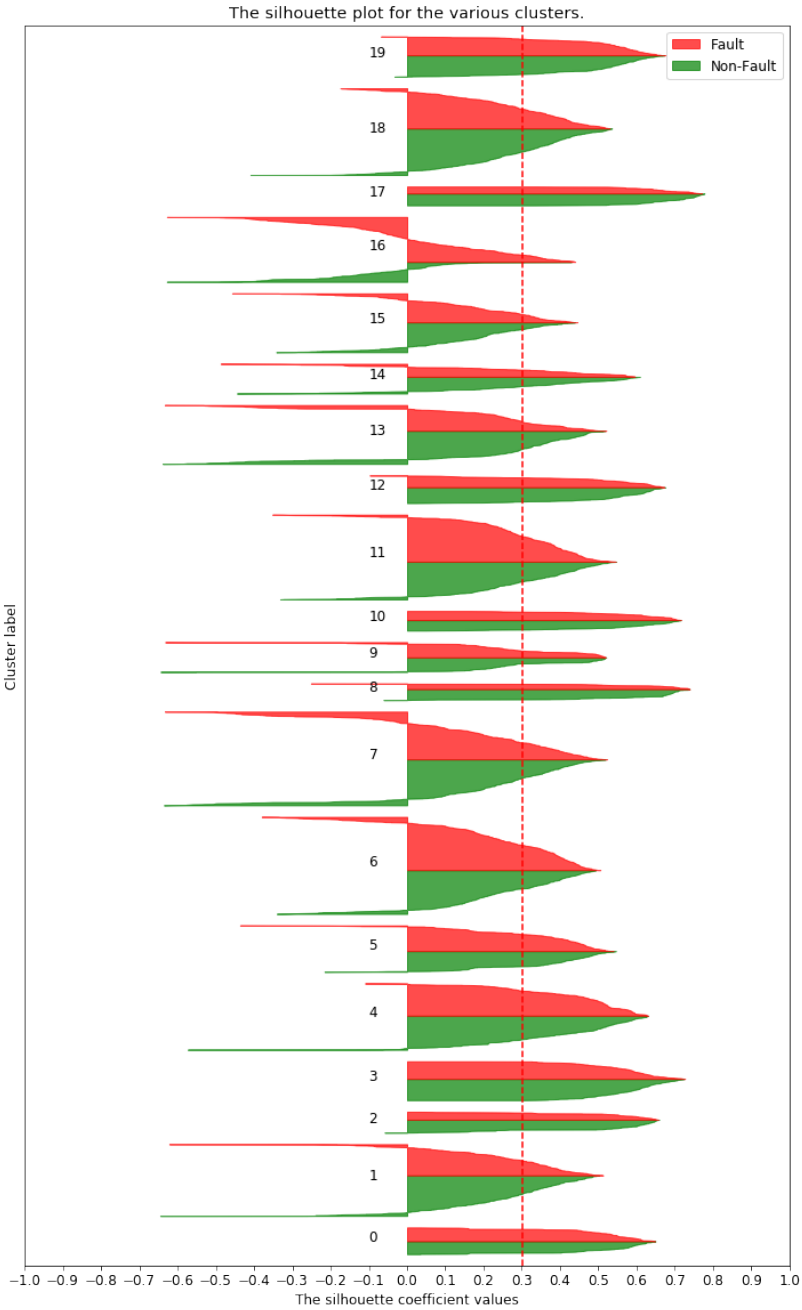


Figure 6.6: Silhouette scores of Gaussian mixture with 20 clusters on normalized and t-SNE reduced data with 3 dimensions on Data set 2.

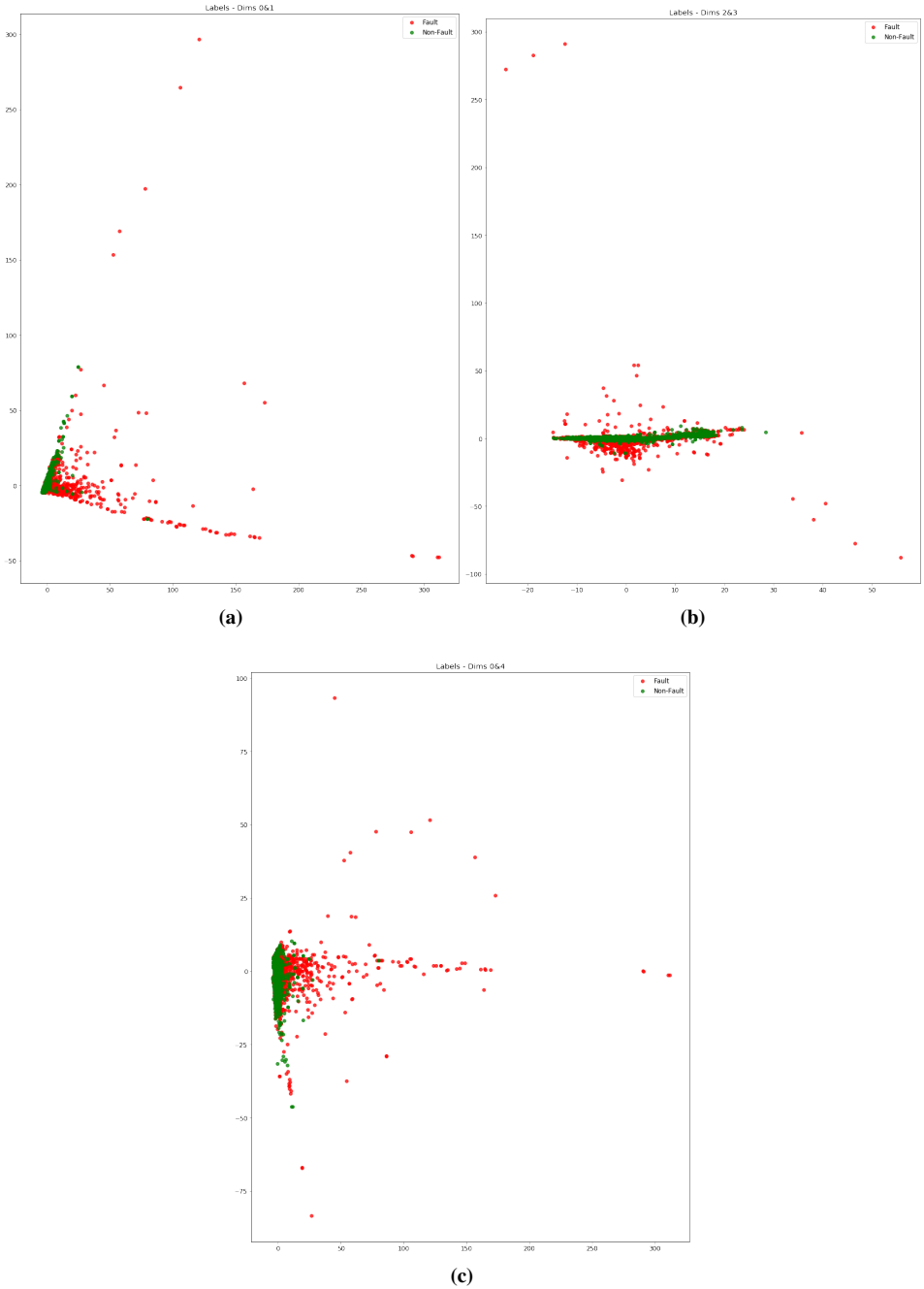
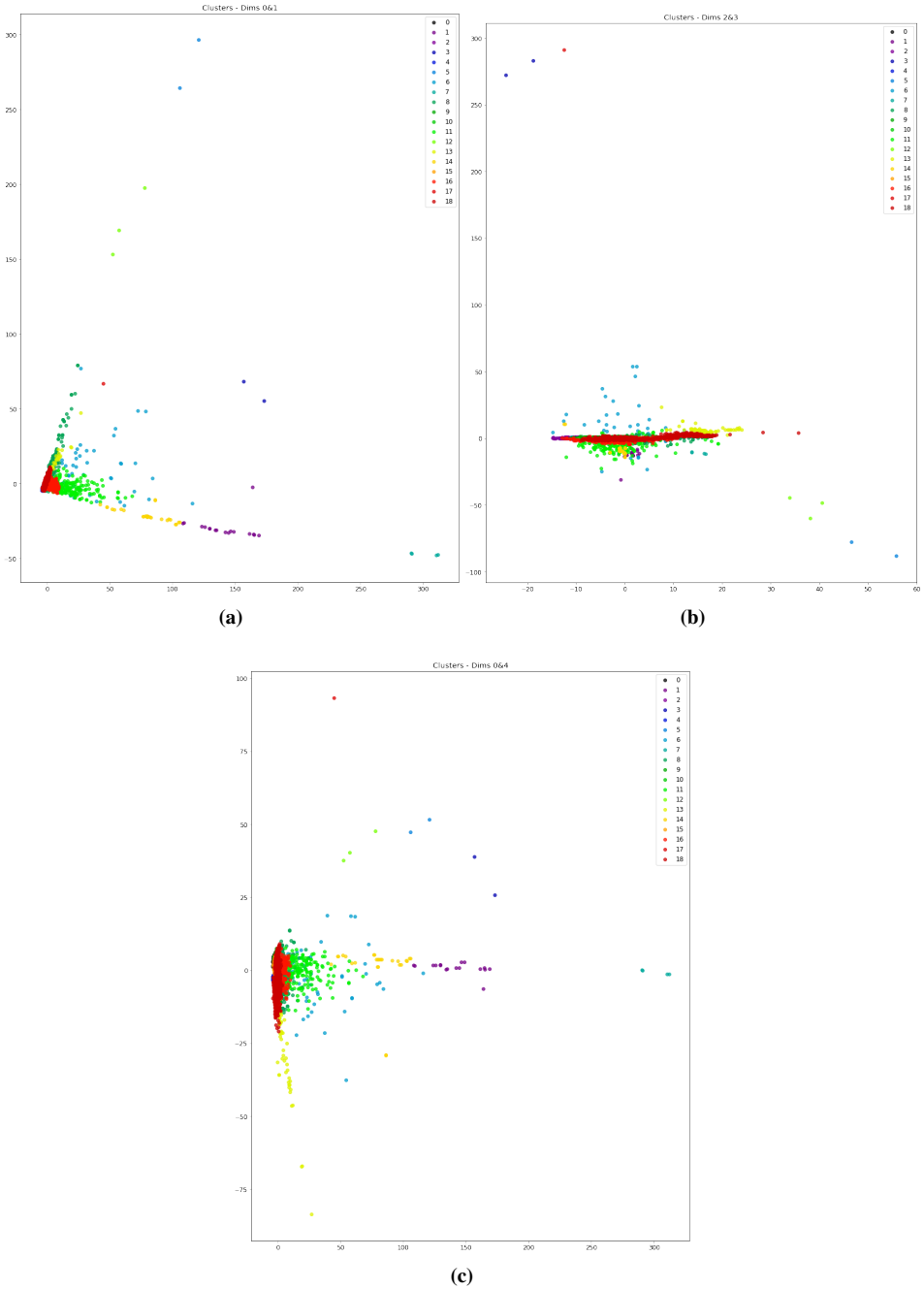


Figure 6.7: Data spread of normalized and PCA reduced data with 50% explaining variance on data set 3.



(a) **(b)**

(c)

Figure 6.8: Clusters of Gaussian mixture with 19 clusters on normalized and PCA reduced data with 50% explaining variance on data set 3.

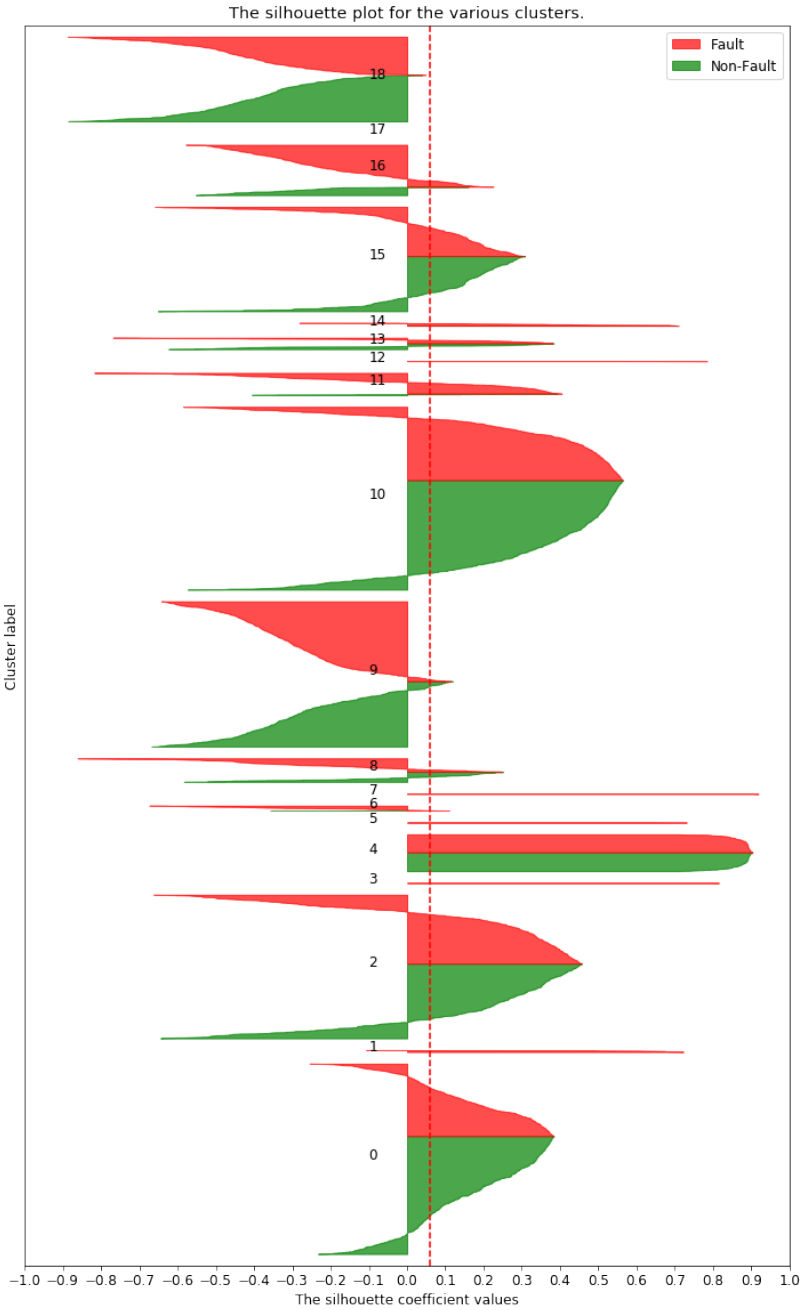
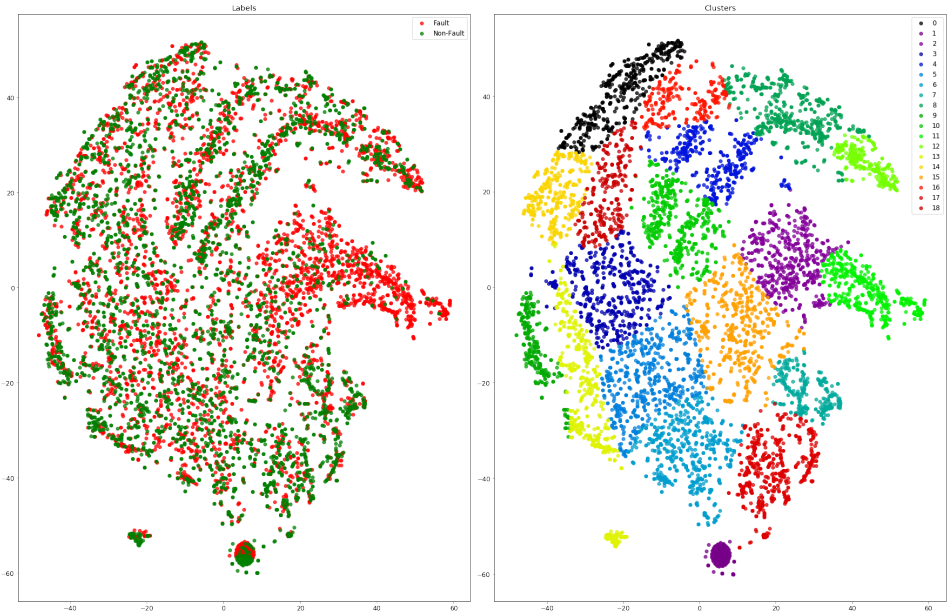


Figure 6.9: Silhouette scores of Gaussian mixture with 19 clusters on normalized and PCA reduced data with 50% explaining variance on data set 3.



(a) **(b)**

Figure 6.10: (a) shows data spread of t-SNE reduced data with 2 dimensions on data set 3+. (b) shows clusters of Gaussian mixture with 19 clusters on t-SNE reduced data with 2 dimensions on data set 3+.

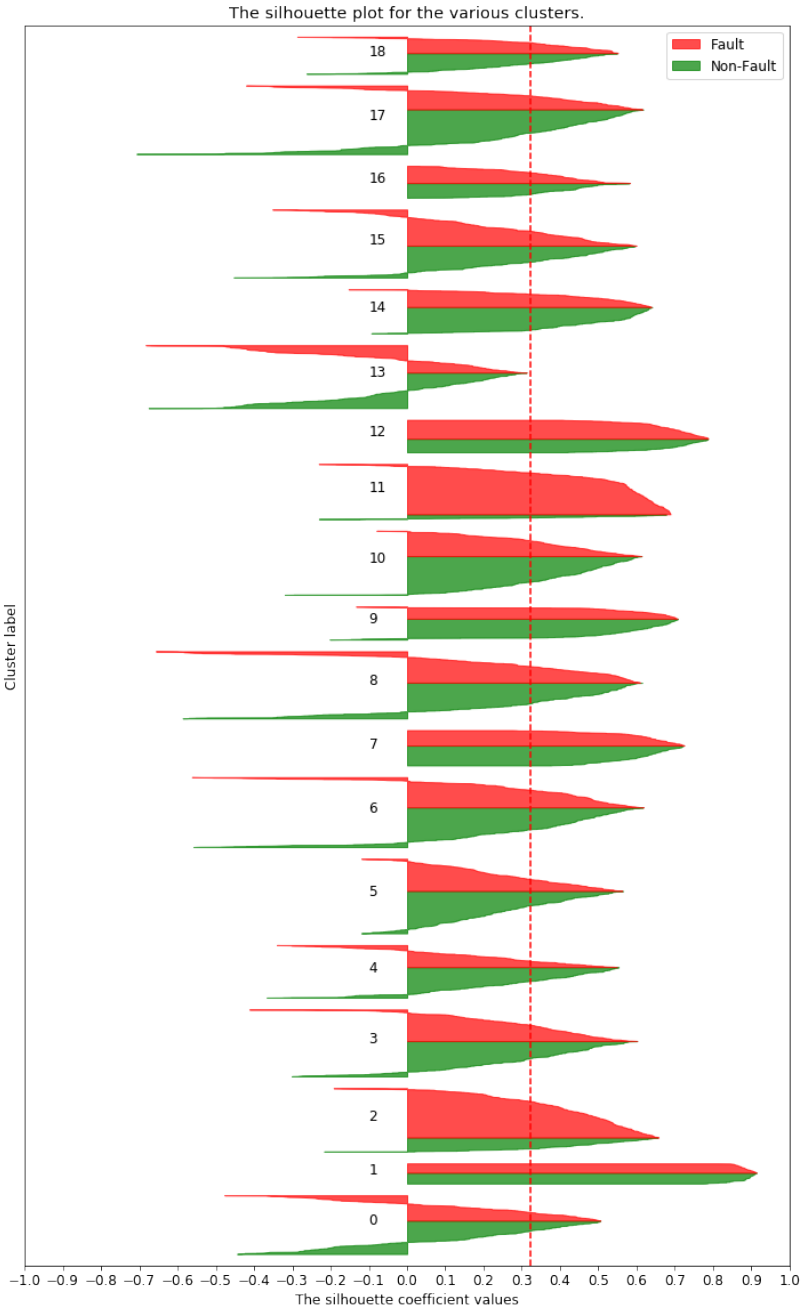


Figure 6.11: Silhouette scores of Gaussian mixture with 19 clusters on t-SNE reduced data with 2 dimensions on data set 3+.

Chapter 7

Future Work

In this chapter we present topics, ideas, and methods which we think could be beneficial to explore in future works regarding the EarlyWarn project.

7.1 Data Sets

7.1.1 Unbalanced Data Sets

Despite the power grid experiencing comparably few faults compared to normal behaviour, we chose to work on a balanced data set. However, this might result in the models getting an incorrect understanding of the data. It might be practical to learn on an unbalanced data set instead. Due to the lack of fault samples, this can also help creating significantly bigger data sets, without having to create artificial data.

7.1.2 Single Fault Type

Some faults might be simpler to differentiate from non-faults. Based on the findings in [Santi, 2019] power interruptions were the easiest to classify while rapids voltage variations were the most difficult. It might be worth looking into creating different models for each fault separately than just one model for all of the faults.

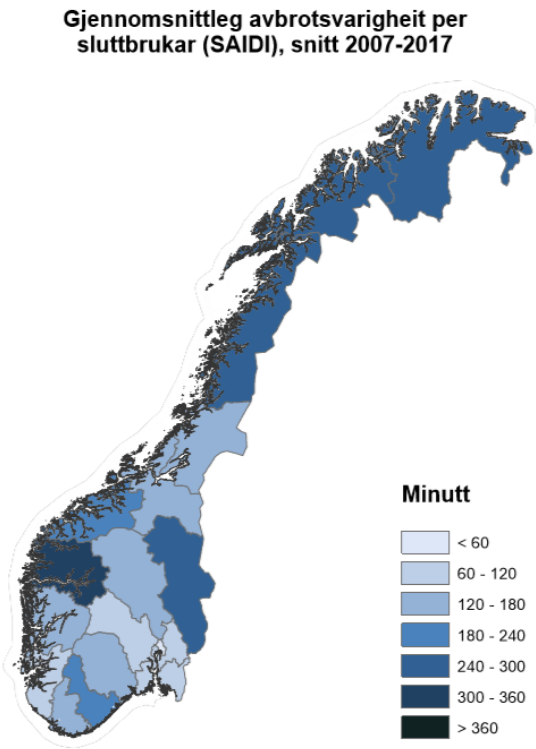


Figure 7.1: Map showing average interruption duration per end consumer from 2007 to 2017 [Norges vassdrags-og energidirektorat, 2019].

7.1.3 Single Location

How the power grid looks during normal behaviour might not be invariant of location. According to [Norges vassdrags-og energidirektorat, 2019] the amount and severity of faults varies largely from county to county as can be seen in Figure 7.1 and Figure 7.2. It might prove beneficial to create models for sets of locations rather than one model for all locations. Some faults might be more apparent at some locations and by creating models for locations that experience similar faults might make it easier for the models to learn the underlying structure.

7.1.4 Separate Only Faults

It might be interesting to inspect the difference solely between faults, as this could hint at what characterizes various faults. This could be used to create better features for the

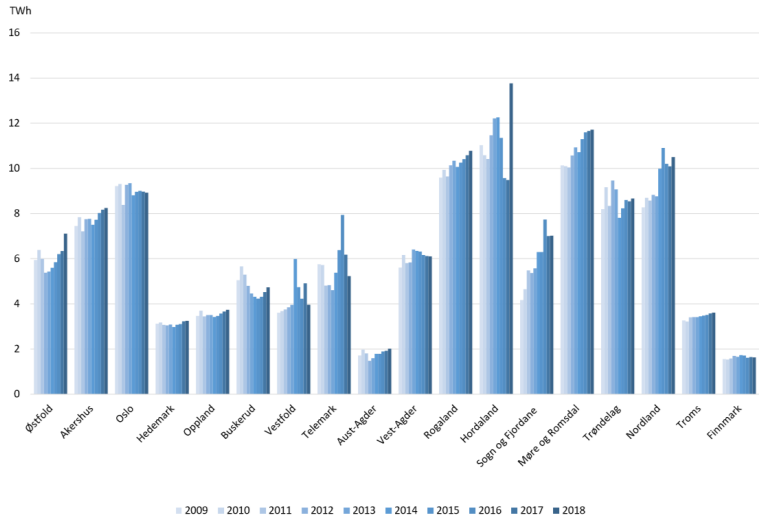


Figure 7.2: Graph showing ILE per county from 2009 to 2018 [Norges vassdrags-og energidirektorat, 2019].

model.

7.2 Features

As presented in Section 6.2, the features used for the experiments in this report, and those used in [Santi, 2019], are the maximum, the mean, and the standard deviation for each harmonic of each phase. However, these features do not retain information about changes over time, nor are they well suited at representing the frequency and duration of irregularities. To better include this information we suggest some additional features.

7.2.1 Additional Aggregated Features

Signal-to-Noise ratio

A feature that indicates the frequency of abnormally large amplitudes might help differentiate samples. Our suggested aggregation method is the signal-to-noise ratio, *SNR*, which can be calculated as $SNR = \frac{\mu}{\sigma}$, where both have the mean – μ – and the standard deviation – σ – already exist as features.

Outlier Count

Rather than calculating the SNR, it might be simpler, and more effective, to calculate outliers, where outliers might be defined as $\lambda \times \mu$, for some λ. This would additionally allow the calculation of both abnormally high amplitudes as well as abnormally low amplitudes by using $\lambda > 1$ and $\lambda < 1$ respectively.

Aggregated Features for Partitions of the Time-Series

Rather than aggregating values for the entire time-series, the time-series could be partitioned into shorter sections. The features can be aggregated for each of these sections. This will better retain information about change over time, and the duration of abnormal behaviour.

7.2.2 Distance Matrix Between Time-Series

If one assumes that each fault emerge in a unique, or some unique fashions, it might be reasonable to calculate the distance between each sample over the entire time-series, where similar time-series would have a lower distance between them. The distances could be used to build a distance matrix, where time-series with low distance could be assumed to be of similar fault-type.

7.2.3 Data ageing

Data ageing is how the relation between newer and older data is managed as mentioned in Section 5.2. It usually is giving less importance to old data and more importance to newer data. As of now the time dimension is ignored completely resulting in equal importance for new and old data. It would be interesting to implement data ageing when introducing the time dimension and look closer at how different importance values affect the results. It could also give insight into how long time before a fault the data starts being relevant, and as such how early it is possible to discover a plausible fault.

7.2.4 Other dimentionality reduction methods

t-SNE showed potential as a dimentionality reduction method, using other probability based dimentionality reduction methods might create good separation of the data.

7.3 Other methods

7.3.1 Clustering

While the clustering methods presented in this report did not create good answers, it is possible, albeit unlikely, that other methods might have better success, perhaps on new dimentionality reduction methods. Possible methods of interest are Spectral Clustering [Ng et al., 2002], DBSCAN [Schubert et al., 2017], OPTICS [Ankerst et al., 1999], and AutoClass [Cheeseman et al., 1988].

7.3.2 Prediction

Expand on [Santi, 2019] by using different machine learning models and by using models that can handle time series data.

(3D) CNNs

It might be interesting to try using Convolutional Neural Networks (CNNs) [S. Albawi and Al-Zawi, 2017] on either spectograms created using Fourier-transform or another method, or on the plots of the original wave. To be able to handle time series data a 3D CNN could be used on multiple spectograms created over a time interval.

LightGBM

As decision-trees already have proven to give good results [Santi, 2019], LightGBM [LightGBM, 2019] might also perform well as it is an improvement on decision-trees. It is also extremely popular on competitions hosted by Kaggle[1] which further speaks for its performance. LightGBM might give deeper insights into which features (harmonics) that are most important and by so shed new light into which harmonics we should use when predicting and clustering.

7.4 Online learning

Thus far have the methods explored not been made to utilize the constant sampling on the power grid. Exploring the feasibility of online learning and finding finding optimal con-

[1] An online community of data scientists and machine learners. https://www.kaggle.com/

figurations for models capable of online learning might prove beneficial as briefly looked into in Section 5.2.

Chapter 8

Conclusion

We conclude by answering the research questions presented in Chapter 1.

RQ1: What is the impact of dimensionality reduction, and how does it affect the clustering results?

As suggested in 4.3.1, dimensionality reduction is essential due to the high dimensionality of the data. t-SNE gives a good basis for clustering, giving good silhouette scores for data set 2 and data set 3+ due to being made for representing neighbourhoods in the original data. The reduced data ends up in a region where distance measures such as euclidean distance is suitable. PCA however did not give a good basis for clustering. The resulting data points were densely packed, with significant outliers being the only thing which could be reasonably separated from the rest.

RQ2: To what extent do there exist underlying structures in the data?

As seen in Figure 6.10(a), most of the data points are in one major neighbourhood, which indicates that there are no greatly separating structural differences between the data points. However, two small neighbourhoods can be seen at the bottom of the figure, and there are some denser areas in the major neighbourhood, as well as some empty areas on the right side, which suggests that there are some structural differences in the data. Even though we did not discover any very apparent structures, this only reflects the result of our chosen features and models. There might be potential structures in the data that may be found using another set of features and different models.

RQ3: How can the structures be used to differentiate between faults and non-faults?

As concluded in Chapter 6 are there some faults which are structurally different from other data when using certain features. These faults can be seen on the right hand side in Figure

6.10(a). This indicates that some faults shares some unique structure. However, we could not find any structural differences between faults and non-faults for the majority of the data.

Bibliography

Aggarwal, C. C., Hinneburg, A., Keim, D. A., 2001. On the Surprising Behavior of Distance Metrics in High Dimensional Space. Springer.

Ankerst, M., Breunig, M. M., Kriegel, H.-P., Sander, J., 1999. Optics: ordering points to identify the clustering structure. In: ACM Sigmod record. Vol. 28. ACM, pp. 49–60.

Bengio, Y., Courville, A., Vincent, P., 2013. Representation learning: A review and new perspectives. IEEE Transactions on Pattern Analysis and Machine Intelligence 35 (8), 1798–1828.

Blackard, J. A., Dean, D. J., 1999. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables.

C. A. Andresen, B. N. Torsaeter, H. H., Uhlen, K., 2018. Fault detection and prediction in smart grids. 9th IEEE International Workshop on Applied Measurements for Power Systems, AMPS 2018 - Proceedings.

C. C. Aggarwal, J. Han, J. W., Yu, P. S., 06 2003. A framework for clustering evolving data streams.

Cheeseman, P., Kelly, J., Self, M., Stutz, J., Taylor, W., Freeman, D., 1988. Autoclass: A bayesian classification system. In: Machine learning proceedings 1988. Elsevier, pp. 54–64.

Davarikia, H., Barati, M., Znidi, F., Iqbal, K., 2018. Real-time integrity indices in power grid: A synchronization coefficient based clustering approach. 2018 IEEE Power Energy Society General Meeting (PESGM).

Domingos, P., 2012. A few useful things to know about machine learning. Communications of the ACM 55, 78–87.

e24, 2018. Ruster opp kraftnettet for milliarder. `https://e24.no/energi/i/gPm10B/ruster-opp-kraftnettet-for-milliarder-et-historisk-hoeyt-nivaa`, accessed: 2019-12-03.

ElspecLTD, 2019. G4400-3-phase class power quality analyzer. https://www.elspec-ltd.com/metering-protection/power-quality-analyzers/g4400-power-quality-analyzer/, accessed: 2019-12-01.

Høiem, K. W., 2019. Predicting fault events in the norwegian electrical power system using deep learning. https://nmbu.brage.unit.no/nmbu-xmlui/handle/11250/2608290?locale-attribute=no, accessed: 2019-11-24.

Kassem, A., Eissa, M., 08 2018. Hierarchical clustering based optimal pmu placement for power system fault observability. Heliyon.

Lichman, M., 2013. Uci machine learning repository.

LightGBM, 2019. Lightgbm. https://lightgbm.readthedocs.io/en/latest/, accessed: 2019-12-10.

Lin, J., Lin, H., Aug 2009. A density-based clustering over evolving heterogeneous data stream. In: 2009 ISECS International Colloquium on Computing, Communication, Control, and Management. Vol. 4. pp. 275–277.

Mansalis, S., Ntoutsi, E., Pelekis, N., Theodoridis, Y., 06 2018. An evaluation of data stream clustering algorithms. Statistical Analysis and Data Mining: The ASA Data Science Journal 11.

Mitchell, T. M., 1997. Machine Learning. McGraw-Hill Science/Engineering/Math.

Miyamoto, S., Ichihashi, H., Honda, K., 2008. Algorithms for Fuzzy Clustering: Methods in c-Means Clustering with Applications. Springer Publishing Company.

Ng, A. Y., Jordan, M. I., Weiss, Y., 2002. On spectral clustering: Analysis and an algorithm. In: Advances in neural information processing systems. pp. 849–856.

Norges vassdrags-og energidirektorat, 2019. Avbrotsstatistikk 2018. http://publikasjoner.nve.no/rapport/2019/rapport2019_29.pdf, accessed: 2019-11-25.

Powers, D., Ailab, 01 2011. Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation. J. Mach. Learn. Technol 2, 2229–3981.

Rousseeuw, P. J., 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Computational and Applied Mathematics 20, 53–65.

Russell, S., Norvig, P., 2016. Artificial Intelligence: A Modern Approach,. Pearson Education Limited.

S. Albawi, T. A. M., Al-Zawi, S., Aug 2017. Understanding of a convolutional neural network, 1–6.

S. Guha, N. Mishra, R. M., O'Callaghan, L., 2000. Clustering data streams, 359–.
URL http://dl.acm.org/citation.cfm?id=795666.796588

Santi, V. M., 2019. Predicting faults in power grids using machine learning methods. https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2625848/no.ntnu%3Ainspera%3A2531236.pdf?sequence=1&isAllowed=y, accessed: 2019-11-24.

Schubert, E., Sander, J., Ester, M., Kriegel, H. P., Xu, X., 2017. Dbscan revisited, revisited: why and how you should (still) use dbscan. ACM Transactions on Database Systems (TODS) 42 (3), 19.

Sentralbyrå, S., 2016. Kraftinvesteringer i støtet. https://www.ssb.no/energi-og-industri/artikler-og-publikasjoner/kraftinvesteringer-i-stotet/, accessed: 2019-12-03.

Sentralbyrå, S., 2019a. Betydelig investeringsoppgang i 2019. https://www.ssb.no/energi-og-industri/artikler-og-publikasjoner/betydelig-investeringsoppgang-i-2019, accessed: 2019-12-03.

Sentralbyrå, S., 2019b. Kraftforsyning bidro til investeringsvekst i 2018. https://www.ssb.no/energi-og-industri/artikler-og-publikasjoner/kraftforsyning-bidro-til-investeringsvekst-i-2018, accessed: 2019-12-03.

Seymour, J., 2001. The seven types of power problems. https://download.schneider-electric.com/files?p_Doc_Ref=SPD_VAVR-5WKLPK_EN, accessed: 2019-11-25.

Silva, C., Saraee, M., 2019. Understanding causes of low voltage (lv) faults in electricity distribution network using association rule mining and text clustering.

Statnett, 2019a. Årsstatistikk 2018. driftsforstyrrelser, feil og planlagte utkoplinger i 1-22 kv-nettet. https://www.statnett.no/contentassets/5fb5605039314f498ed16f8561695a0c/arsstatistikk-2018-1-22-kv.pdf, accessed: 2019-11-25.

Statnett, 2019b. Årsstatistikk 2018. driftsforstyrrelser og feil i 33-420 kv-nettet. https://www.statnett.no/contentassets/5fb5605039314f498ed16f8561695a0c/arsstatistikk-2018-33-420-kv.pdf, accessed: 2019-11-25.

Vadlamudi, V. V., 2018. Fundamentals of power systems refresher for tet4115 (power system analysis). N/A.

van der Maaten, L., Hinton, G., 2008. Visualizing data using t-sne. Journal of Machine Learning Research 9, 2579–2605.

Zweig, M. H., Campbell, G., 1993. Receiver-operating characteristic (roc) plots: a fundamental evaluation tool in clinical medicine. Clinical Chemistry 39, 561–577.