Master's thesis

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

Halvor Kvernes Meen and Camilla Jahr

Power Wave Analysis and Prediction of Faults in the Norwegian Power Grid

Master's thesis in Computer Science
Supervisor: Helge Langseth
June 2020

NTNU
Norwegian University of Science and Technology

SINTEF

Halvor Kvernes Meen and Camilla Jahr

Power Wave Analysis and Prediction of Faults in the Norwegian Power Grid

Master's thesis in Computer Science
Supervisor: Helge Langseth
June 2020

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

NTNU
Norwegian University of
Science and Technology

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 thesis we will do a thorough analysis of the data obtained from the Norwegian Power grid, and try to find out to what extent it is possible to use this data to predict faults in the power grid. We present different ways of representing the data, and different machine learning methods suitable for prediction. We then look at the different data representations to see if there are any noticeable differences between the structures in the faults and the non-faults, and if so what might have caused these differences. We finally use the machine learning methods to try to predict that a fault will occur within different time intervals and forecast horizons.

We discover that using the raw waveform instead of other popular representations such as the Fourier transform gives the best results. We also find that using a signal with a very high resolution does not necessarily improve the performance, but that it is more important to look at the signal over larger time intervals. Lastly we discover that there are some differences in the structures in the data, but they are mainly caused by their origin nodes and not whether it is a fault or not. Looking at each node separately, the differences between the structures in the faults and non-faults become a bit more visible.

Keywords Norwegian Power Grid, Power Analysis, Fault Prediction, Machine Learning

Sammendrag

Det moderne samfunnet har blitt avhengig av elektrisitet, og som følger av dette har strømnettet blitt en viktig del av infrastrukturen vår. Å tilby et stabilt kraftdistribusjonsnett er ekstremt viktig og sørger for at både industrien og husstander kan ha en forutsigbar kilde til energi. Med fremskrittene til maskinlæring og lagringskapasitet av store data har det oppstått et ønske om å kunne forutse feil på det forfallende strømnettet slik at man kan sikre stabilitet for brukerne.

I denne masteroppgaven skal vi gjøre en gjennomgående analyse av data fått fra det norske strømnettet, og prøve å finne ut til hvilken grad det er mulig å bruke denne dataen til å predikere feil i strømnettet. Vi presenterer ulike måter å representere dataen på, og ulike maskinlæringsmetoder passende for prediksjon. Deretter ser vi på de ulike datarepresentasjonene for å se om det er noen merkbare forandringer i strukturen til feil og ikke-feil, og om så hva som kan være årsaken til disse forandringene. Til slutt bruker vi maskinlæringsmetodene til å prøve å predikere om en feil kommer til å inntreffe innenfor ulike tidsintervaller og ulike tider før feilen eventuelt inntreffer.

Vi oppdager at å bruke den opprinnelige bølgeformen istedet for andre populære representasjoner som Fourier transformasjonen gir de beste resultatene. Vi finner også ut at å bruke et signal med veldig høy oppløsning ikke nødvendigvis forbedrer resultatene, men at det er viktigere å se på signalet over større tidsintervaller. Til slutt oppdager vi at det er noen forskjeller i strukturene i dataen, men at dette hovedsaklig er forårsaket av hvilke noder dataen stammer fra, og ikke om det er en feil eller ikke. Hvis man ser på hver node individuelt blir forskjellene mellom strukturene i feil og ikke-feil litt mer tydelige.

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 thesis. 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, Bendik Torsæter, Volker Hoffmann and Torfinn Tyvold. Thank you for clarifying the project and for providing us with the resources we needed. Especially thanks to Christian Andresen for the many helpful meetings and support, and to Volker Hoffmann for dealing with us and all the server problems we caused, and for giving a lot guidance and insight into the mysterious world of wavelets.

Table of Contents

List of Tables

List of Figures

Abbreviations

AC	=	Alternating Current
A-HA	=	Automatisk Hendelsesanalyse (Automatical Incident Analysis)
AUC	=	Area Under the Curve
A	=	Current
CNN	=	Convolutional Neural Network
CPU	=	Central Processing Unit
DC	=	Direct Current
DDG	=	Dynamic Data-set Generator
EM	=	Expectation Maximization
FN	=	False Negative
FP	=	False Positive
GPU	=	Graphics Processing Unit
GMM	=	Gaussian Mixture Model
ILE	=	Ikke Levert Energi (Not Delivered Energy)
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
STD	=	Standard Deviation
STFT	=	Short Time Fourier Transform
TN	=	True Negative
TP	=	True Positive
t-SNE	=	t-distributed Stochastic Neighbor Embedding
V	=	Voltage

Chapter 1

Introduction

This thesis is a part of the EarlyWarn project. The main purpose of EarlyWarn is to develop surveillance systems that predicts and identifies faults and disturbances in the Norwegian power grid. EarlyWarn is presented in more detail in Chapter 3.

This Master's thesis is a continuation from the work done in the specialization project [Jahr and Meen, 2019]. As the background is the same, parts of the introduction (Chapter 1) and the background (Chapters 2, 3, and 4) in this thesis will be based on the corresponding chapters in the specialization project.

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 during 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 [Statistisk Sentralbyrå, 2016]. In 2019 all the investments totalled to about 40 billion NOK which was a small downfall from 2018, but seen in a historic perspective, it is still a considerable sum [Statistisk Sentralbyrå, 2019a]. Number of investments for the last years

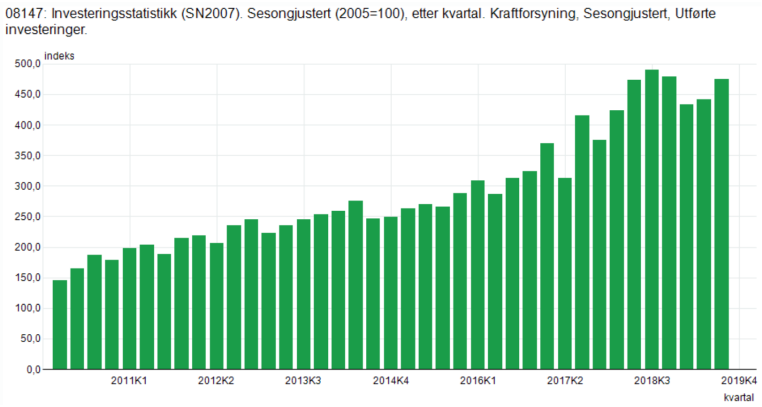


Figure 1.1: Statistics for the period 2011-2019 showing the number of investments adjusted after seasons (Statistisk Sentralbyrå).

can be seen in Figure 1.1. The Norwegian industry has also had a steady increase in energy consumption over the last years [Statistisk 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 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 advancements 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 thesis is to do a thorough analysis of the wave signal data obtained through the EarlyWarn project, and to find out to what extent it is possible to use this data to predict faults in the power grid. We want to explore whether or not there are structures in the data prior to faults occurring that can be used for prediction, and if there are other factors than characteristics of the faults that affect the wave signal, and are apparent in the data, for instance seasonal and geographical differences. To achieve these goals, the following research questions have been formulated:

- **RQ1:** To what extent do there exist differentiable structures in the data?
- **RQ2:** Which data representations are the most useful for predicting faults in the power grid?
- **RQ3:** How long before faults occur does the signal contain information which differentiates them from normal behavior?
- **RQ4:** What prediction performances are achievable using machine learning methods?

Chapter 2

Background - Power Grids

Parts of this chapter are based on the specialization project [Jahr and Meen, 2019], with some added methods and analysis in Section 2.3.

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 cause 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 have 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

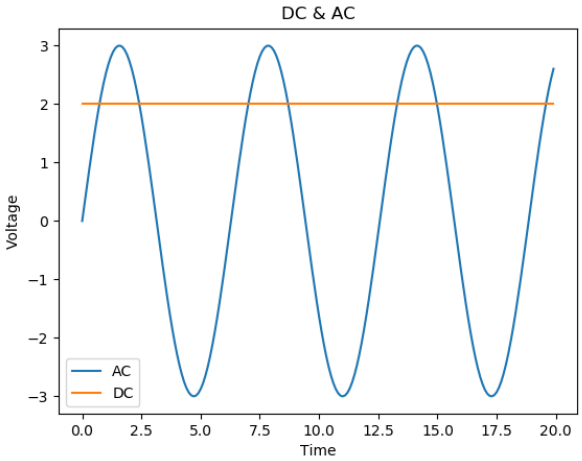


Figure 2.1: Example of a direct- and an alternating current.

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.

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:

$$e^{it} = \cos t + i \sin t \tag{2.2}$$

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

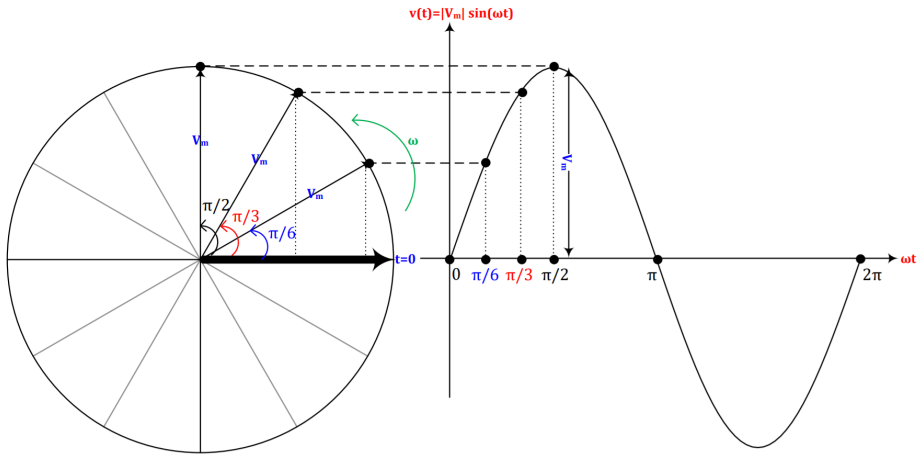


Figure 2.2: The relationship between a phasor and sinusoidal wave [Vadlamudi, 2018].

where e is Euler's number and i is the imaginary unit, we can rewrite Equation 2.1 to [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 explained earlier, 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).

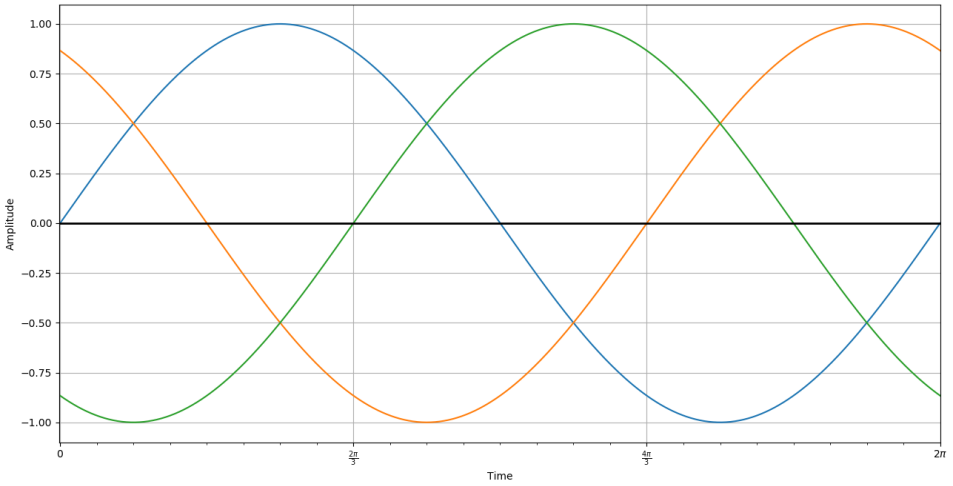


Figure 2.3: Sinusoidal wave representation of three phase power with $\frac{2}{3}\pi$ radians as phase offset.

To build a three phase generator three coils are placed $\frac{2}{3}\pi$ radians (120°) apart (See Figure 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 are 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.

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)*

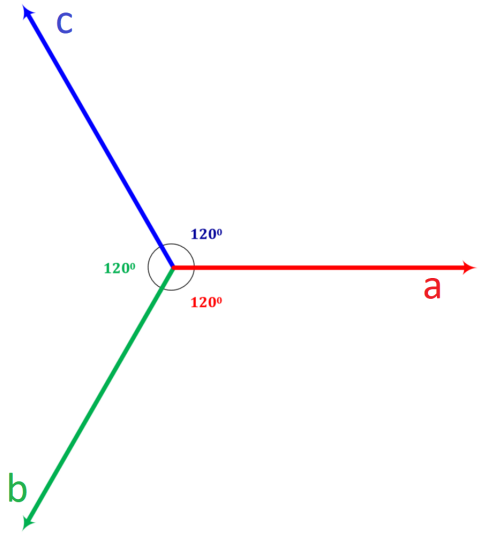


Figure 2.4: Phasor diagram representation of three phase power, phases a, b and c, with $\frac{2}{3}\pi$ radians (120°) as phase offset.

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 are not always enough to determine if there has been an error and might require further investigation.

2.3 Analysis

There are many different ways of looking at and representing the wave of a power signal. Furthermore there are just as many methods for retrieving valuable information from these representations. Now we will look at and compare some popular ways of representing the wave of a power signal.

2.3.1 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 $\hat{f}$:

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

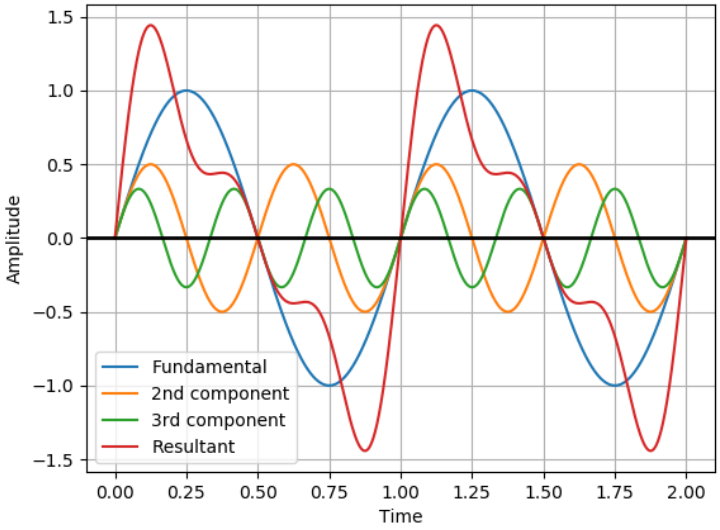
Discrete Fourier Transform

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:

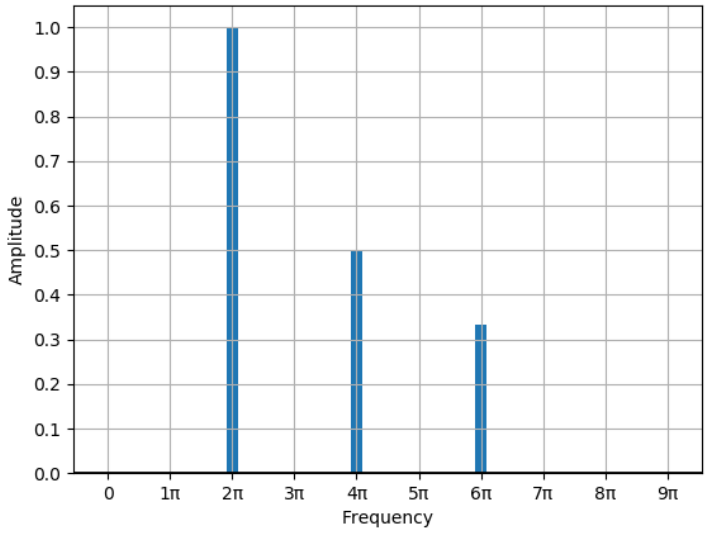
$$X_k = \sum_{n=0}^{N-1} x_n e^{-\frac{2\pi i}{N} k n}$$

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

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



(a)



(b)

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

By using Euler's formula (Equation 2.2) with $t = \frac{2\pi}{N}kn$ we can rewrite the discrete Fourier transform as:

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

Short Time Fourier Transform

A disadvantage of the Fourier transform is that it removes all information about changes in regards to time. Short time Fourier transform (STFT) addresses this by reintroducing the time domain. Explained simply, STFT divides the wave of the signal into equal-sized segments and then computes the Fourier transform over each segment separately. By doing this one can observe the changes in frequencies from one segment to another. The STFT can easily be derived from either the Continious- or the discrete Fourier transform by multiplying with a windowing function:

$$STFT\{x(t)\}(\tau,\omega) = \int_{-\infty}^{\infty} x(t)w(t-\tau)e^{-i\omega t}dt$$

where $x(t)$ is the signal, $w(\tau)$ is the windowing function and ω is the frequency (continuous-time STFT). The discrete-time STFT is further derived by changing the continuous signal $x(t)$ with a discrete version $x[n]$ and the continuous time value for the windowing function τ with a discrete time value m:

$$STFT\{x[n]\}(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]w(n-m)e^{-i\omega n}$$

One of the main drawbacks STFT has is that is has a fixed resolution, the width of the windowing function that segments the wave of the signal is constant and cannot be varied. As such one must take a compromise between frequency resolution and time resolution as illustrated in Figure 2.6. Frequency resolution describes how easy it is to tell apart components with frequencies that are close to each other, similarly time resolution describes how easy it is to see at which times the frequencies change. A wide window gives a low time resolution, but a high frequency resolution and vice versa. As illustrated in Figure 2.7 the STFT with a narrow window makes it easy to see at which points in time the frequencies are, but the frequencies themselves are blurry. The wide window is opposite, it is easy to see the frequencies, but it is not clear at which points in time they occur.

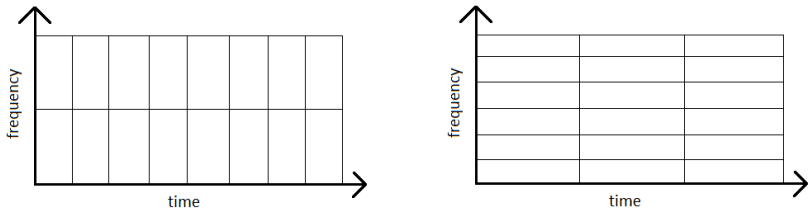


Figure 2.6: Comparison of STFT window size. Left with small window sizes giving better time resolution, and right with bigger window sizes giving better frequency resolution.

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 frequencies 2π, 4π and 6π, and amplitudes 1, $\frac{1}{2}$ and $\frac{1}{3}$ respectively. Figure 2.5(b) shows the Fourier coefficients, the frequencies, with the belonging amplitudes.

2.3.2 Harmonics

In regards to electric power systems, harmonics are multiples 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 2.8 for a visual representation).

The distorted sinusoidal can be decomposed by using the discrete Fourier transform, resulting in an 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.

2.3.3 Wavelet Transform

Wavelet transform is very similar to STFT in the sense that it tries to fit a number of functions to a given segmented signal. The difference being that while the STFT tries to

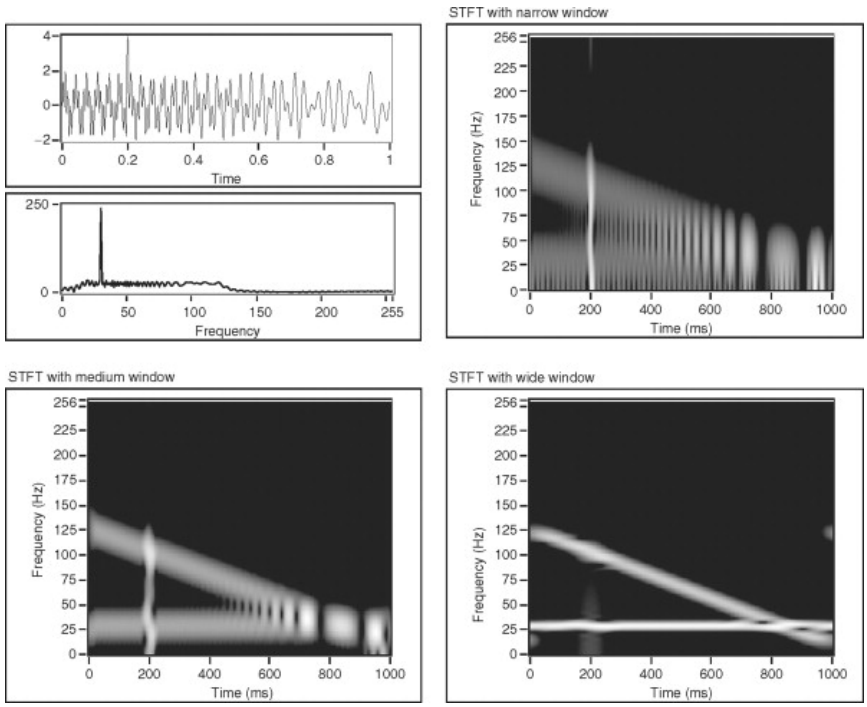


Figure 2.7: STFT spectograms of a signal with different window sizes [Kehtarnavaz, 2008].

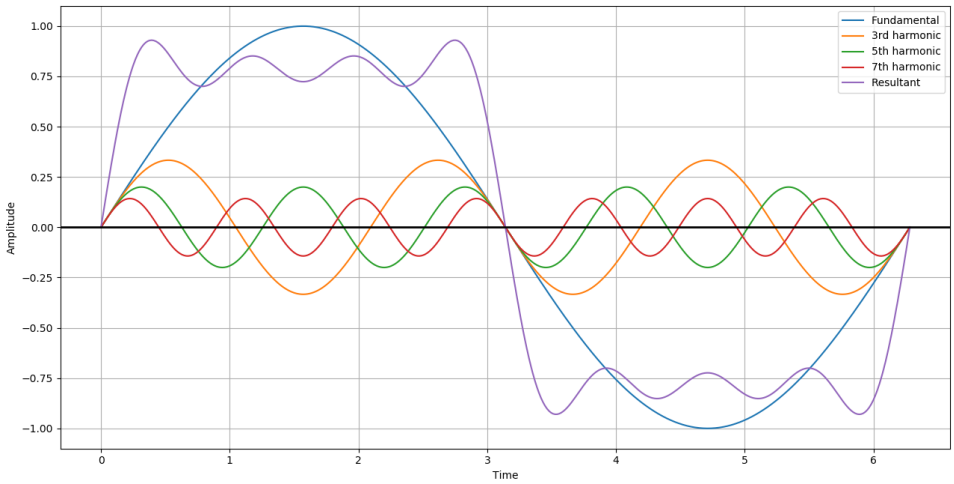


Figure 2.8: Resultant of the 1st, 3rd, 5th and 7th harmonic.

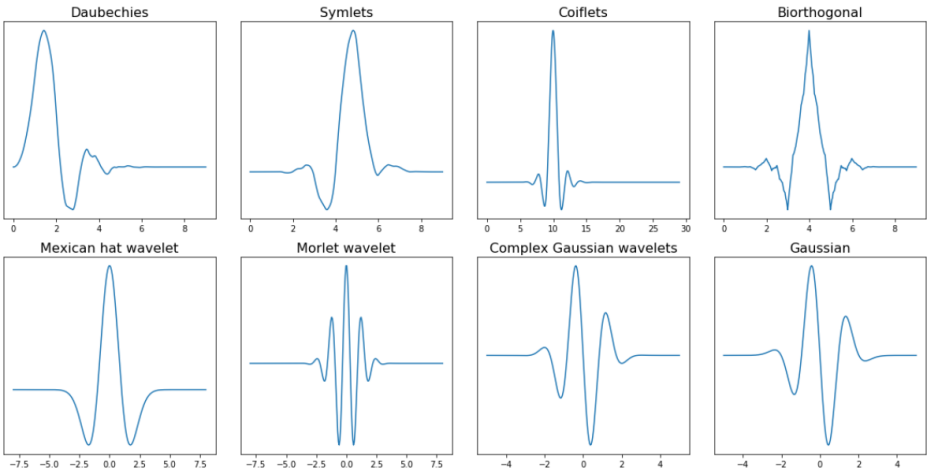


Figure 2.9: Some common wavelet families. There are also multiple variations within each family.

fit infinitely many sine-functions with a fixed window size, the wavelet transform tries to fit wavelets. Wavelets are wave-like oscillations that are characterized by their amplitudes starting and ending with 0, as well as the mean being 0. There are very many different wavelets for different usages, some of the most common are shown in Figure 2.9. Wavelets are defined with **scaling** and **shifting**[3]. The scale is related to the window length for the STFT (See Section 2.3.1) and describes the size of the wavelet. The scale is inversely proportional to the frequency. A higher scale helps to capture the slowly varying changes of the signal, while a lower scale helps to capture more sudden and abrupt changes. The shifting describes where in time the wavelet is located. An illustration of the resolutions of the wavelet transform is shown in Figure 2.10. By varying the scale and shift it is possible to get a representation that captures both sudden and slow changes over the entire signal. This means it is possible to both have a high frequency resolution for small frequency values as well as high time resolution for large frequency values. In other words, at scales where we are interested in features dependent on time we can choose a high time resolution and at scales where we are interested in features dependent on frequency we can choose a high frequency resolution.

As with Fourier there are both a continuous transform and a discrete transform. Continuous wavelet transform lets scaling and shifting vary continuously, giving potentially infinitely many wavelets. It is expressed by the following integral:

$$X_\omega(a,b) = \frac{1}{|a|^{1/2}} \int_{-\infty}^{\infty} x(t)\psi(\frac{t-b}{a})dt$$

where ψ is the complex conjugate of a given wavelet, a is the scale and b is the shift. Discrete wavelet transform has discrete scaling and shifting. The scale increases in powers

[3]Many also calls shifting for translation. These are interchangeable, but we will stick to the term **shifting**.

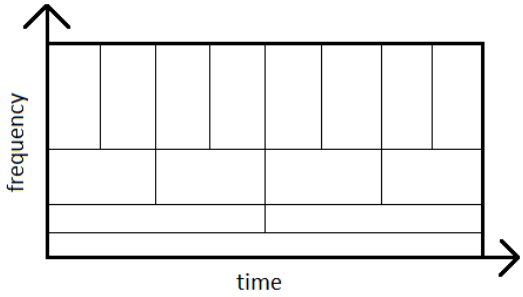


Figure 2.10: Illustration of the time and frequency resolution of the wavelet transform.

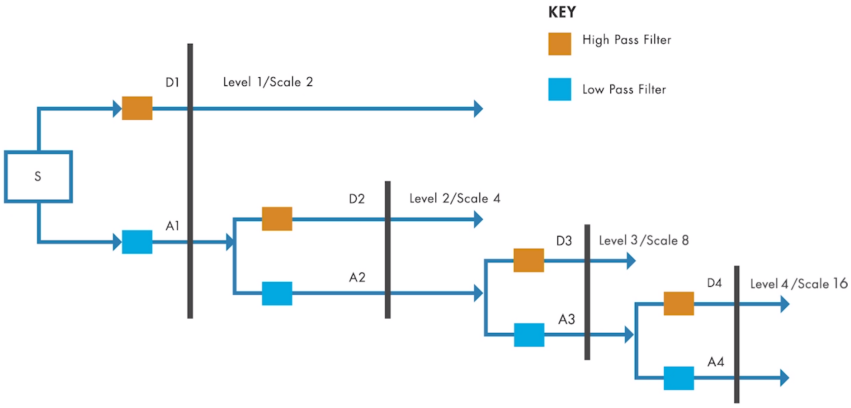


Figure 2.11: Illustration of the decomposition of the discrete wavelet transform [Devleker, 2016]. **D** is the coefficients from the high pass filters which together makes up the returned output. **A** is the coefficients from the low pass filters that are sent down for further decomposition. At each level the scale is multiplied by two and the number of samples are halved.

of two (1, 2, 4, 8..) and the shift is integer values (1, 2, 3, 4..). Discrete wavelet transform decomposes the signal through filter banks, the signal is passed through a cascade of high pass and low pass filters. At each level of the filter bank the signal is decomposed into high and low frequencies as shown in Figure 2.11 and the scale increases by a factor of two (meaning that the frequency decreases with a factor of two). As half of the frequencies are removed, half of the samples can be discarded as per the Nyquist Theorem[4] reducing the computational cost. This is continued until all desired frequencies are captured or there are no more samples left. The coefficients from the high pass filter are returned while the coefficients from the low pass filters are sent to the next level where the process is repeated.

[4]The Nyquist Theorem states that for a given signal the sampling rate should be twice as large as the frequency of its highest frequency component.

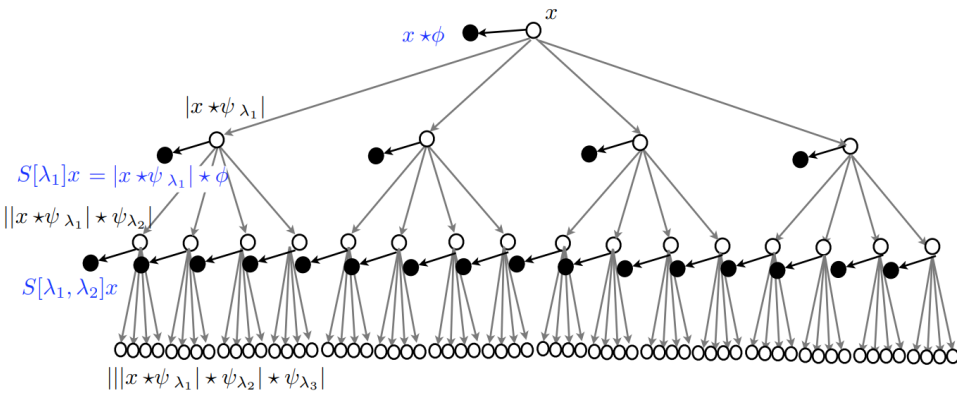


Figure 2.12: Illustration of wavelet scattering [Mallat, 2012]. Here x is the signal, ψ is the wavelet and ϕ is the averaging operator. The $\star$ is the convolution operator, where a convolution is like an inner product. As seen the signal is decomposed using wavelets which coefficients are being used the modulus operator on. At each layer the averaging operator is used to calculate the value S which together makes up the final returned output.

Wavelet Scattering

Wavelet scattering works in a similar manner to the cascading filter banks used in discrete wavelet transform. The signal is first decomposed through a low pass filter and a high pass filter. The output from the high pass filter is then again decomposed in the same way, and this is repeated creating a layered network as shown in Figure 2.12. High scale wavelets are used as low pass filters as they capture the low frequencies, and low scale wavelets are used as high pass filters. It is possible to create as many layers as one desires, but in practice it is enough with three as the energy dissipates at every iteration making sure that all the energy of the wave is captured in the last layer. The coefficients that are outputted from the low pass filters are averaged over, giving one coefficient for every set of shifts (with a given window size) for each scale. The averages from each layer in the network are given as the output. The wavelet scattering network is very similar to the convolutional neural network explained in Section 4.4.5, with wavelets being the already learned filters which do not need training, and averaging as the pooling function.

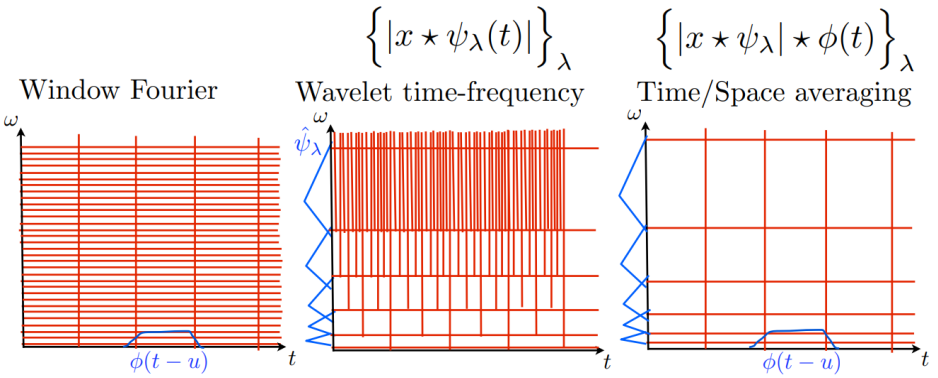
The first layer simply gives wavelet coefficients extracted from each frequency band. As these have been made only using information from a low pass filter they do not contain information from higher frequencies. The output from the second layer, and the layers further down, contains information about higher frequencies as it is based on the outputs from the first high pass filter. As the second layer uses wavelet transform on the outputs from the first layer which also are wavelet coefficients, it is not obvious what it outputs. As each wavelet isolates a band of frequencies, the wavelet transforms in the second layer further isolate frequencies in the frequency bands given from the first layer. This can be thought of as measuring the interferences/differences between the frequencies in the

frequency bands. The third layer will then find the interferences of the interferences and the fourth layer interferences of interferences of interferences and so forth.

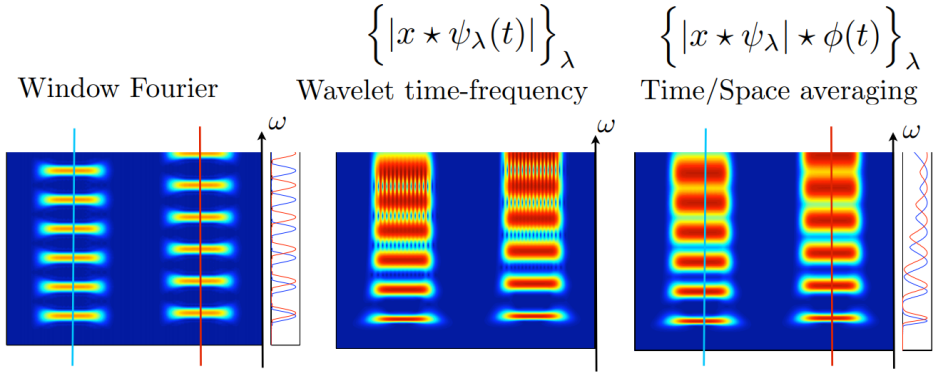
2.3.4 Comparison

One of the reasons wavelets are preferred over Fourier as a signal representation is because Fourier is not **stable** at high frequencies. A method being stable means that if there is a small deformation in the signal, we expect the transformation to have a change in the same order as the deformation (linear). This is important as signals with small deformations might look the same to the human eye, we might perceive them as the same ”class”, but the spectograms given by an unstable transform might represent them totally different. A deformation can for instance be a change in the frequency or in the amplitude of the signal. The Fourier transform is unstable at high frequencies meaning small deformations in the frequencies of the signal give big differences in the transformation even though we expect them to be close. A problem also arises for wavelets at the higher frequencies, as in order to capture the high frequencies the wavelet is scaled down resulting in a high time resolution. This makes it sensitive to changes in shifting as it is highly localized in time. There are multiple methods for removing this sensitiveness, one of the most common being taking the average over the coefficients at the cost of resolution. See Figures 2.13 and 2.14 for illustrations.

We do not want to lose resolution, and this is where wavelet scattering comes to the rescue. By averaging over the low pass filters it gets invariant (stable) to local time shifting, in addition to using high pass filters in order to retain the information lost in the low pass filters (keeping the frequency resolution), see Figure 2.15 for an illustration. This does however not come free and requires more processing power and storage, and one must decide if stability at higher frequencies and time localized info are worth the added cost.



(a)



(b)

Figure 2.13: (a) and (b) show the windowed Fourier, wavelet transform, and wavelet transform with time averaging [Mallat, 2012]. (a) shows the difference in resolution between the three and (b) shows the spectograms of the three transforms used on a signal and a slightly deformed signal. ψ is the wavelet at different scales λ showed by the blue lines on the y-axis. ϕ is the averaging operator done at different shifts $(t-u)$ showed by the blue lines on the x-axis. In (a) you can see that resolution is sacrificed when averaging, and in (b) that this sacrifice gives a more stable transformation. On the right side of the spectograms the coefficients are plotted as curves for a point in time for both the original signal and the deformed one, shown by a blue and red line respectively. The curves for the Windowed Fourier are stable at lower frequencies, but as the frequency increases it is apparent that waves get more and more different. The oscillations in the coefficients from the wavelet scattering have disappeared in the time averaged wavelet scattering due to the averaging, and as a result the plotted curves are much more similar for higher frequencies.

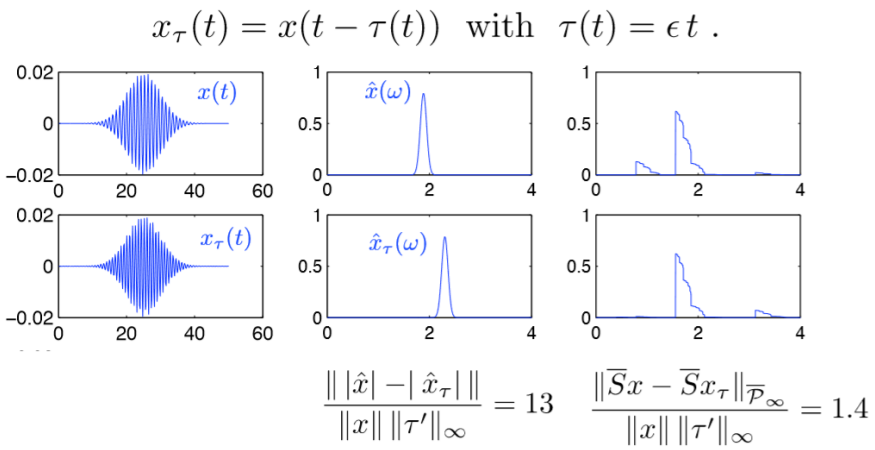


Figure 2.14: Illustration of stability [Mallat, 2012]. The first column shows a signal x twice, with the bottom one being slightly dilated. The middle column shows the Fourier transform, and as you can see the frequency support of the dilated signal has moved to the right. If you were to calculate the distance by subtracting it from the original signal, it would be considerably large relative to the deformation. The last column shows the wavelet scattering transform, and if you were to calculate the distance now, it would be a lot smaller as the frequency support has not moved.

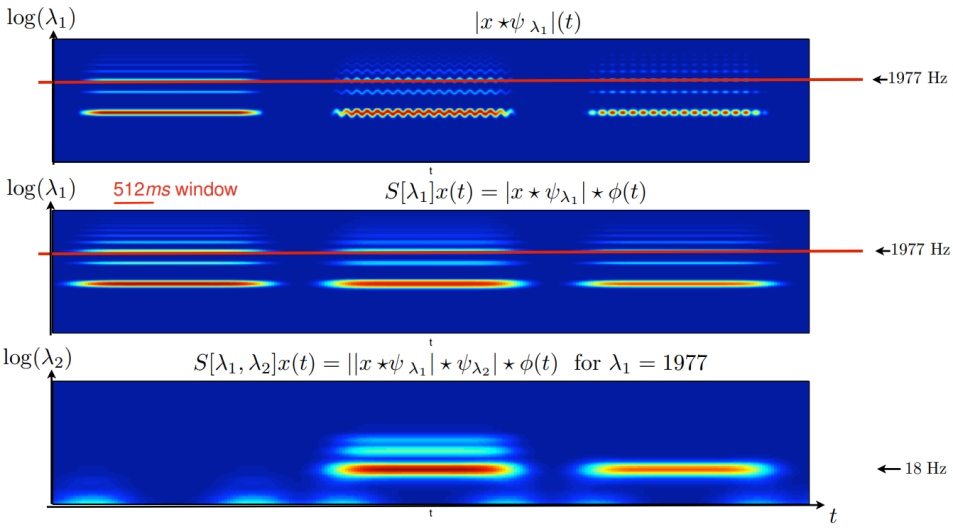


Figure 2.15: Illustration of three signals and their wavelet scattering coefficients for the first and second layer with the two bottom spectograms being averaged over time [Mallat, 2012]. The y-axis shows the frequencies (given by the index of the wavelet scale) and the x-axis shows time. The red line shows the frequency that the bottom spectogram is made from. The three different signals differ mostly in the higher frequencies, and because of that they look almost completely the same in the first layer when averaged. However, all the inner structure and information about the higher frequencies are preserved in the second layer and they are easy to tell apart. In the spectogram of the second layer, most of the energy is at 18Hz. This implies that the most apparent interference frequency given $log(\lambda_1)$=1977Hz is at 18Hz.

2.4 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 were a lot more than normal, but only 740 operational faults on the transmission- and regional net, which were very few 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*[5] has made a categorization utilized in the annual reports and can be viewed in Figure 2.16 in context of operational faults, and in Figure 2.17 in context of undelivered power (ILE[6]). As can be seen in the figures, *surroundings* are the biggest cause of both operational faults and ILE. The surroundings were further categorized into subcategories as can be seen in Figure 2.18 and Figure 2.19. 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.

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

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

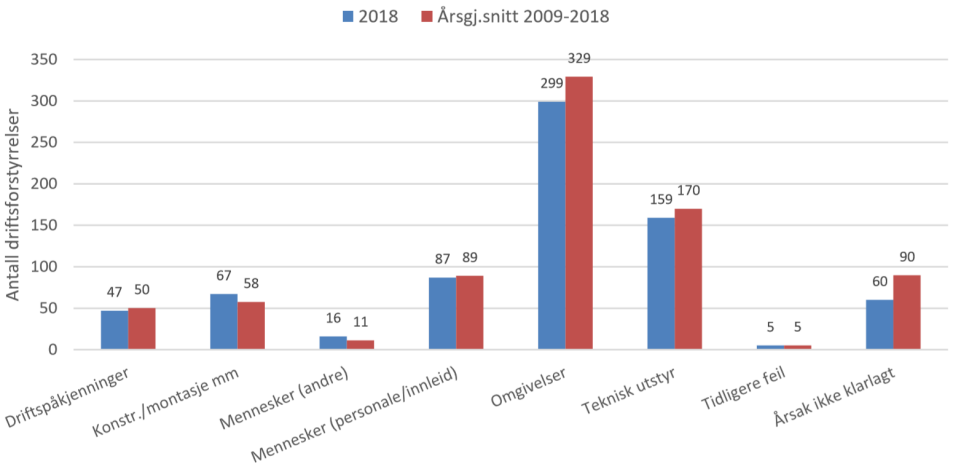


Figure 2.16: Overview of operational faults on the transmission- and regional net and their causes [Statnett, 2019b].

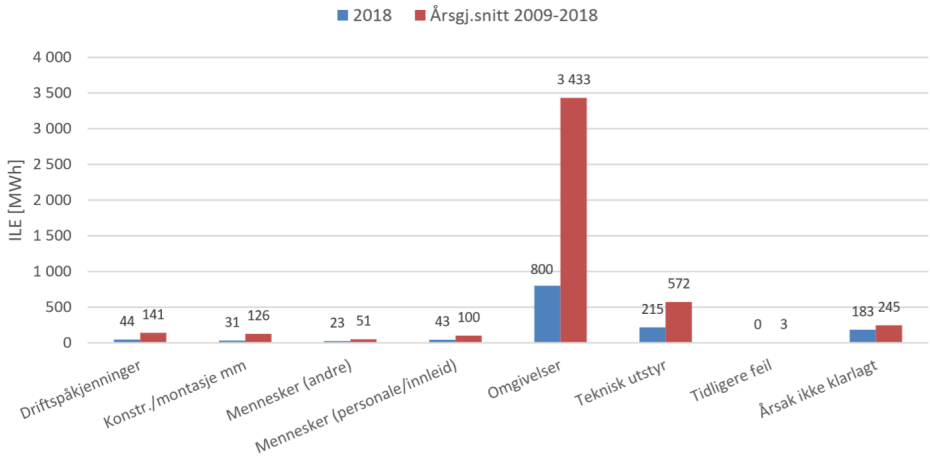


Figure 2.17: Overview of ILE on the transmission- and regional net and their causes [Statnett, 2019b].

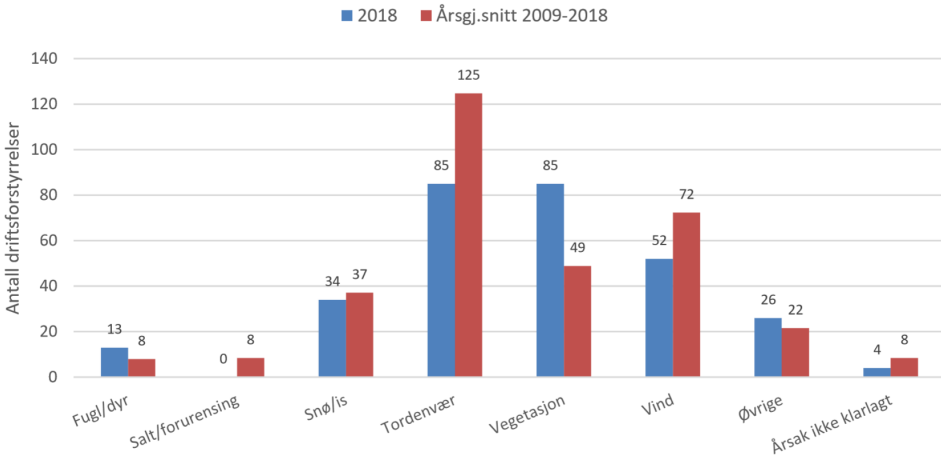


Figure 2.18: Overview of operational faults on the transmission- and regional net caused by surroundings [Statnett, 2019b].

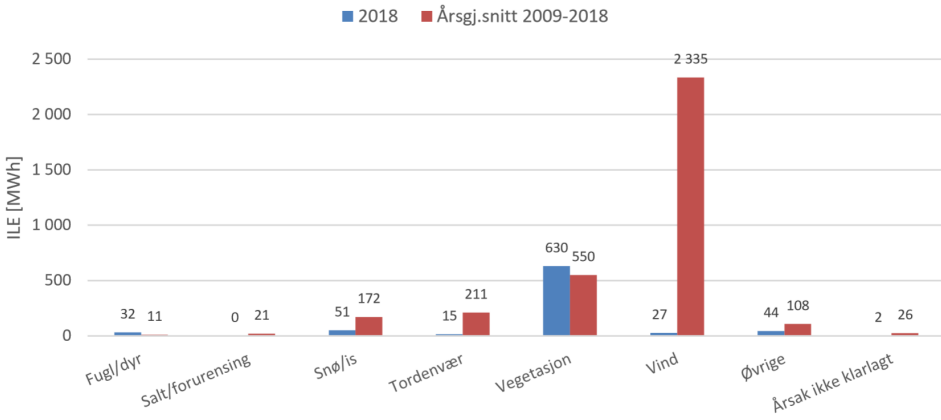


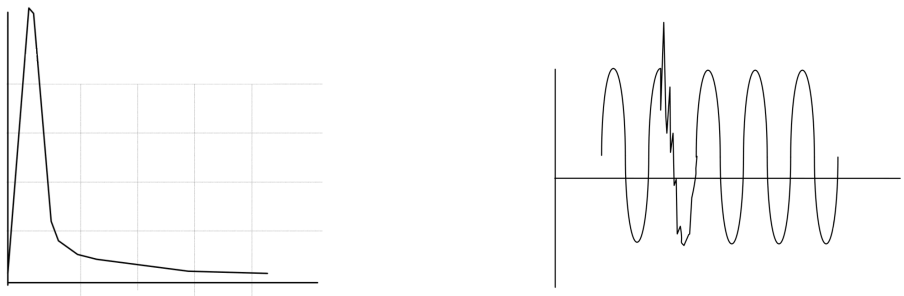
Figure 2.19: Overview of ILE on the transmission- and regional net caused by surroundings [Statnett, 2019b].

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 thesis. 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 2.20); impulsive and oscillatory transients. Impulsive transients are the most common type of power surge/spike and involves a sudden increase or decrease of the voltage/current level. They usually span a very short time interval. Causes include lightning, grounding failure and equipment faults to name a few. Oscillatory transients cause 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 2.21) 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 2.20: Example of transients [Seymour, 2001].

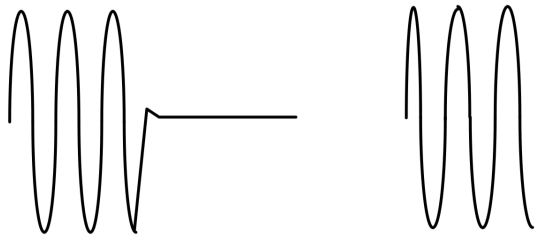


Figure 2.21: Example of a momentary interruption [Seymour, 2001].

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

Figure 2.22: 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 2.22a) 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 deliver enough power. Undervoltages (See Figure 2.22b) 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.

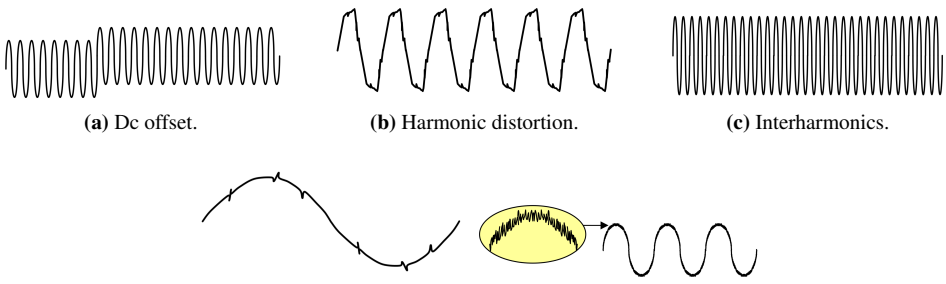
Swell / Overvoltage. A swell (See Figure 2.23a) is the opposite of a sag, that is to say an increase in the voltage that lasts for 0.5 cycles up to a minute. Causes can for instance be the shutdown of equipment that consumes large amounts of power, or faulty isolation. Overvoltages (See Figure 2.23b) 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 affects 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 2.24a) is an offset that results in the average of the wave not being

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

Figure 2.23: Examples of swell and overvoltage [Seymour, 2001].



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

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

Figure 2.24: Examples of waveform distortions [Seymour, 2001].

zero, increasing or decreasing the RMS value 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 2.24b) 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 2.24c) 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 2.24d) 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 2.24e) 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.

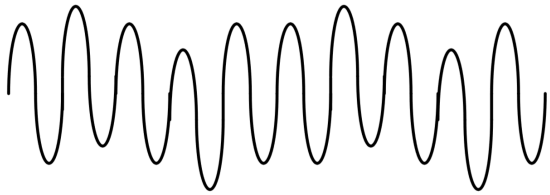


Figure 2.25: Example of a voltage fluctuation [Seymour, 2001].

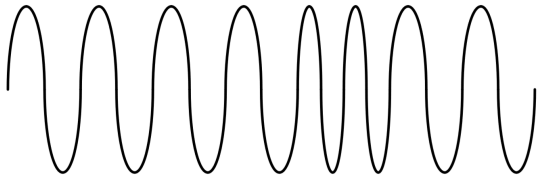


Figure 2.26: Example of a frequency variation [Seymour, 2001].

Voltage fluctuations are series of minor, random changes in the wave of the voltage (See Figure 2.25). The variations are 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 2.26). 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

Parts of this chapter are based on the specialization project [Jahr and Meen, 2019].

In this chapter we will introduce the EarlyWarn project that this thesis 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

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

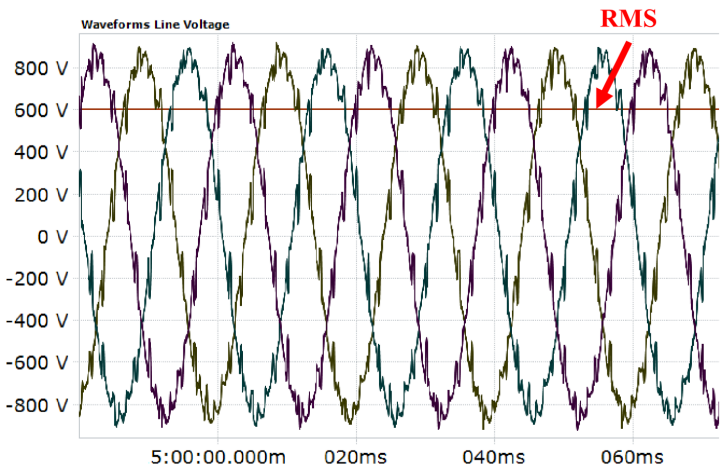


Figure 3.1: Example of a RMS value with its wave affected by harmonic distortion sampled by a PQA [Andresen et al., 2018].

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 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 [Andresen et al., 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 which 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 2.4. PMUs on the other hand provide 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 [Andresen et al., 2018] and are therefore very useful for comparing signals at

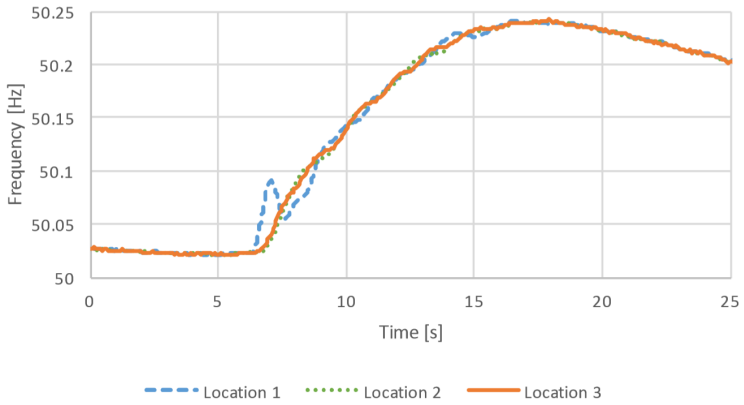


Figure 3.2: Example of frequencies from three locations sampled by a PMU. The frequency suddenly increases at Location 1 because of a loss of load [Andresen et al., 2018].

different locations and monitoring at the transmission-net level (See Figure 3.2).

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 Data-sets

To extract time series from the centralised server, SINTEF made an application called *Dynamic Data-set 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.

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 data-set with both faults and non-faults, the DDG is also able to generate non-fault time series at a ratio given by the user. There is also metadata for each observation in the data-set (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 is true. Similarly a false positive occurs when the system predicts that something is true, but in reality 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 have been 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.

Parameter	Description
Total duration	Time duration to include in the observation
Resolution	Sampling frequency of the signal in the generated observation
Time before fault	Time window between the generated observation and the fault
Data type	What data type to produce. Can choose between the *wave form*, the *Fourier coefficients*, and the *RMS values*
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*
Specificity	Which lines and phases to produce. Can be *V1*, *V2*, and *V3* for the respective phases and *V12*, *V23*, and *V13* for the respective lines
Overlap period	The overlap period used for the data-set. Overlap period is defined in Section 7.1.1

Table 3.1: Parameters for the DDG.

Parameter	Description
Fault detection	Fault or non-fault
Fault type	Fault type, if any
Fault time	Time of occurrence
Start time	Start time of samples
End time	End time of samples
Total duration (sec)	Seconds of data in the data-set
Resolution (ms)	Time interval between each sample
Number of samples	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 observation.

Chapter 4

Background - Machine Learning

Parts of this chapter are based on the specialization project [Jahr and Meen, 2019].

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 learn-

ing 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 generalizes 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 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 are 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 are 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 default 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.

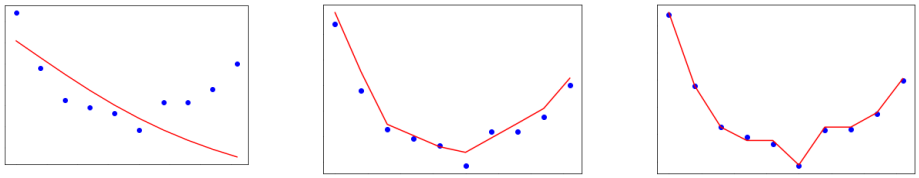
4.2.3 Training, Validation and Testing

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

[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.



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

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

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.

- **Increasing the 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.

(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.

- **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.

- **Dropout** helps generalizing the model in multiple ways. Dropout works by randomly dropping nodes in a neural network, if a node is dropped it will not give any output and is ignored. This helps reducing overfitting in the way that it forces nodes to create new connections and to adapt as the layer structure is constantly changing. This prevents nodes to get too reliant on some of its input and encourages nodes to use all of their inputs. Also, by dropping out some of the nodes the model gets simpler as the capacity is reduced, resulting in a even lower chance of overfitting.

- **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 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. This 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 option 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 represent 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.

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 which are present in the original data, in the chosen dimensions [van der Maaten and Hinton, 2008].

4.3 Ensemble Learning

Ensemble learning is a method where multiple models are used, which together – usually – obtain a better predictive performance than any of the models would by themselves. Ensemble learning makes use of the way supervised models attempt to find the best prediction hypothesis for the data, in a given hypothesis space. By combining multiple prediction hypotheses into a single hypothesis, the ensemble model will – usually – find a hypothesis which better suits the given data.

Ensemble learning is typically used to compensate for simple and poor learning models, by assuring that the output model has done significant calculation on the problem. For instance, random forest, which will be explained further down this chapter, combines hundreds to thousands of decision trees, to obtain a well calculated output.

There are many forms of ensemble learning, in this thesis we will look at two, bagging and boosting.

4.3.1 Bagging

Bootstrap aggregating, usually abbreviated to bagging, is an ensemble learning method where multiple models are used to together determine the results of the ensemble model. When given a data-set X with n samples, a bagging model which aims to use m models will generate m new training-sets X_i, each containing n' samples. X_i is created by sampling from X uniformly and with replacement, causing some samples to be represented multiple times. For large values of n and with $n' = n$, each set X_i is expected to contain roughly $1 - \frac{1}{e} \approx 63.2\%$ unique samples. All of the m models are then fitted using their own training-set, and majority vote decides the output of the ensemble model, giving all the models the same predictive weight. Compared to the basic models the ensemble model will usually have less variance in its decisions, better accuracy, and be significantly less susceptible to overfitting.

4.3.2 Boosting

Boosting is an ensemble learning method where a series of models is used, boosting the results of the prior model. The first model takes in the data-set X, where all samples have equal weight. Once this model is done, the next model takes in the data-set X again, but samples which got correctly categorized by the initial model are given a lower weight, while samples which got incorrectly categorized are given a higher weight. Thus the new learner focuses on correctly classifying the previously incorrectly samples. Compared to the basic models the ensemble model will usually have less bias and variance in its decisions, however it is prone to overfitting. The accuracy yielded by boosting is comparable to, and in some cases better than, the accuracy yielded by bagging.

4.4 Machine Learning Methods

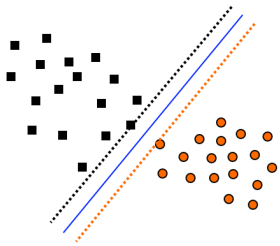
There exists many methods which solve classification problems, here we present the ones used in this thesis.

4.4.1 Support Vector Machines

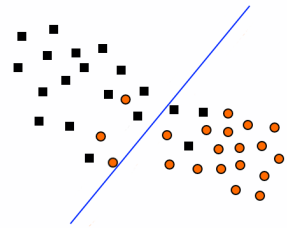
Support Vector Machines, hence called SVMs, are supervised models which perform linear classification and regression analysis. SVMs try to separate labels in a given data-set by linearly separating data with some number of hyperplanes, efficiently making subspaces in the data-space containing the various separated communities of data-points. These hyperplanes are commonly referred to as support vectors. An example of such a hyperplane in a two-dimensional space is given in Figure 4.3.

SVMs might be made to use either a hard or soft margin, meaning that the communities contain *only* one label, or *majorly* one label, respectively. To generalize the subspaces in hard margin SVMs as much as possible, each hyperplane is positioned as far as possible from the closest point in each bordering subspace, making the gap – the margin – between the communities as large as possible. For generalizing soft margin SVMs, the hyperplanes are positioned so that data-points in subspaces consisting of majorly another label are as close to the hyperplanes as possible.

SVMs can be modified to perform non-linear classification by using the kernel trick to alter the data-space. By altering the data-space into a higher dimension, data which is not separable by linear functions in the original dimension can be separated by higher dimensional hyperplanes.



(a) A hard margin SVM. The blue line is the hyperplane separating the two subspaces, and the dotted lines denote the margin.



(b) A soft margin SVM. The blue line is the hyperplane separating the two subspaces.

Figure 4.3: Two Support Vector Machines in $\mathbb{R}^2$.

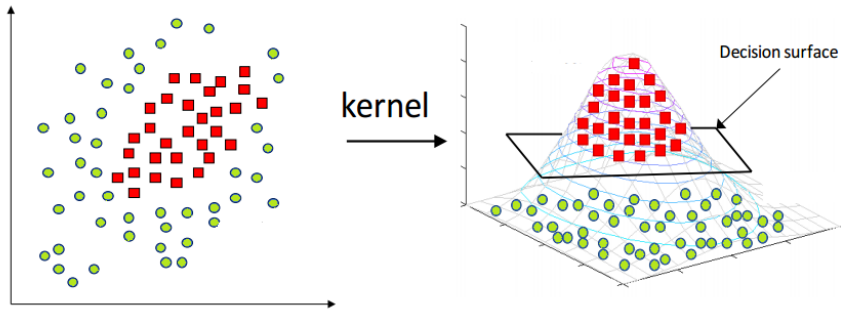


Figure 4.4: The Gaussian kernel applied to a non-linearly separable data-set in $\mathbb{R}^2$, but separable by a hyperplane in $\mathbb{R}^3$. [Sharma, 2019].

Kernel Trick

The kernel trick is a method which utilizes the kernel methods to enable algorithms to work in a high dimensional, feature space, without having to compute the coordinates of the data in that feature space. Rather, the data-points are compared by computing the inner product – used to calculate the distance between two points in the target feature space – between the images of all pairs of data-points. This method is computationally cheaper, and takes less memory, than transforming all points into the target space and then calculating their inner products. The kernel trick can be applied to any linear model, making it non-linear. SVMs are one such example, as illustrated in Figure 4.4.

The kernel trick can mathematically be explained as:

> For each pair of data-points $\mathbf{x}$ and $\mathbf{x}'$ in some input space $\mathcal{X}$, calculate the inner product $k(\mathbf{x}, \mathbf{x}')$ where $k\colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ in some other space $\mathcal{V}$. For some problems it is simpler to write the kernel as a "feature map", where $\varphi\colon \mathcal{X} \to \mathcal{V}$ and $k(\mathbf{x}, \mathbf{x}') = \langle \varphi(\mathbf{x}), \varphi(\mathbf{x}') \rangle_{\mathcal{V}}$, where $\langle \cdot, \cdot \rangle_{\mathcal{V}}$ is a proper inner product. There is no requirement for φ to have an explicit representation.

4.4.2 *k*-Nearest Neighbors

k-nearest neighbor, hence called *k*-NN, is a supervised classification algorithm where each new data-point is labeled similarly to the majority of its *k* closest labeled neighbours. An example of this labeling scheme is shown in Figure 4.5. The closest neighbours are defined as the data-points which have the shortest distance from the data-point, where distance, denoted as $D(x, y)$, has the following properties:

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

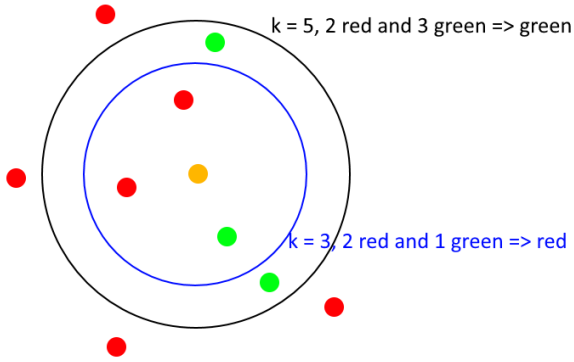


Figure 4.5: k-NN with $k = 3$ and $k = 5$, labeling the new yellow data-point as red for $k = 3$, and green for $k = 5$.

- $D(x, y) = D(y, x)$
- $D(x, y) \leq D(x, z) + D(z, y)$

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 $y = (y_1, ..., y_P)$.

4.4.3 Decision Trees

Decision trees are supervised classification models which work by inferring rules from a labeled data-set as to best predict what label a new, unlabeled, data-point ought to be. The tree is made by splitting the data-set into smaller subsets by evaluating some values in the input data, so that the *information gained* from the split is as high as possible. This is done by finding the split which causes the greatest reduction in *entropy*, where entropy, denoted H, is defined as

$$H(X) = -\sum_{i=1}^{l} p_i \times log_b p_i$$

where X is the data-set, l is the number of unique labels in the data, and p_i is the relative frequency of class i in X. b is the base of the logarithm, and is commonly 2.

The information gained, denoted IG, from splitting on some attribute α can then be defined as

$$IG(X, \alpha) = H(X) - \sum_{v \in values(\alpha)} \frac{|S_\alpha(v)|}{|S|} \times H(S_\alpha(v))$$

where $S_\alpha(v)$ denotes the subset created when choosing samples where $\alpha = v$.

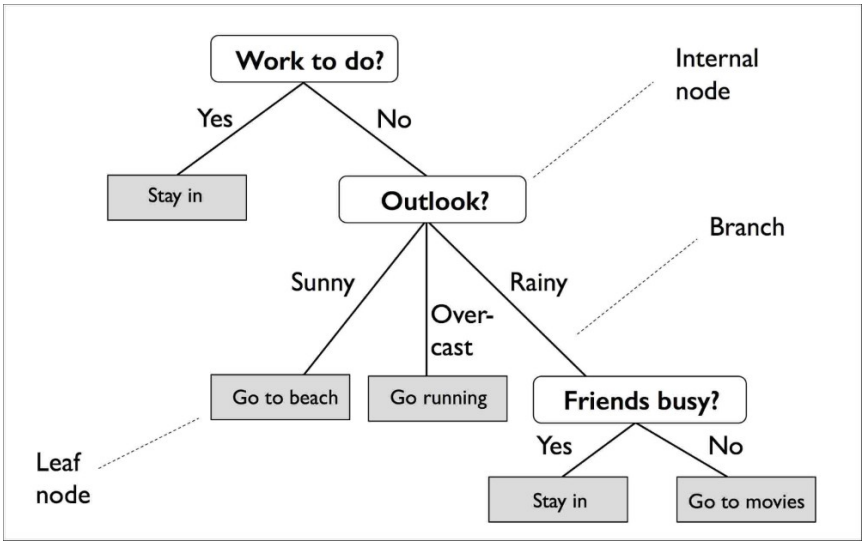


Figure 4.6: A decision tree to determine what a person ought to do on a given day, based on decisions taken about the features, "Work to do", "Outlook" and "Friends busy" [Li, 2019].

Decision trees may be naïvely created by splitting the data-set as dictated by information gain until rules which perfectly categorize each sample in the data-set is made. However, this approach will often lead to very big trees, unique rules to remove outliers, noise, and incorrectly labeled data. In other words, it will be overfitted and fail to classify new data correctly. The common approach to prevent this in decision trees is to (1) limit the depth of the tree, (2) introduce a least information gain requirement to perform a split, and (3) prune the tree. Figure 4.6 illustrated how a decision tree might look.

Random Forest

Random forest is a bagging ensemble learning model consisting of hundreds to thousands of decision trees. The decision trees are all created equally, and function as described above, except for what features they receive as input data. The only way random forest differentiates from general bagging models is that it chooses n' so that X_i contain roughly $\sqrt{p}$ features where p is the amount of features in X. This is done so that if there is a feature p_j which strongly determines the label of the sample, it will not be present in a majority of the decision trees, as those trees would then become too correlated for the random forest to predict well for samples not well describable by p_j.

CatBoost

CatBoost, named by combining the words ”Category” and ”Boosting”, is a boosting ensemble learning library. It is built up of *oblivious* decision trees, meaning that the same decision is used to split the data-set for for each level in the decision tree. This makes it so that the tree only has to calculate one decision for each level, rather than 2^d decisions per level, where the root level is defined as $d = 0$. This greatly increases the speed of which the tree may be made, as well as being less prone to overfitting, at the cost of accuracy [Chepenko, 2019].

4.4.4 Neural Network

Neural network, or NN, is perhaps the most famous term and probably the first thing that comes to mind when it comes to artificial intelligence. It is based on how the brain works and how the cells within the brain communicate with each other. There do exist unsupervised neural networks, but in this thesis we will mainly focus on neural networks as supervised classification models. A neural network usually consists of an input layer, some number of hidden layers, and an output layer at the end. Each layer has a number of nodes with each node having a weight. The relationship between the layers depends on the type of neural network. The first and simplest type of neural network was the **feedforward neural network**, or FFNN, where the connections between the nodes cannot form a cycle. The flow of information only moves one way, from one layer to the next. See Figure 4.7 for an illustration. In a FFNN each node in a layer is connected to each node in the following layer, this is called **fully connected layers**. The number of hidden layers are arbitrary, but the more hidden layers there are in the network, the more complex problems can be solved. However, there is an extra storage and computation cost for each hidden layer. Neural networks with many hidden layers are called **deep neural networks**. The use and application of hidden layers has become its own field of research, a subset of machine learning called **deep learning**.

Each node in a FFNN has some inputs and an output. The inputs are given from the nodes in the previous layer (for the first layer, the input layer, the input will be the input provided by the user) and the output is calculated from the following formula:

$$Z = \sigma(\sum_{i=0}^{n} w_i x_i + b)$$

where Z is the output, w_i is the weight on the input x_i from node i, b is the bias, and σ is the activation function. The activation function is usually a non-linear function that defines the final output of the node. See Figure 4.8 for some common activation functions. The output(s) from the node(s) in the output layer gives the prediction of the network. There is a node in the output layer for each class we want to classify.

Before the network can be used it has to be trained. In the training process the trainable pa-

rameters of the network are tuned in order to minimize a given loss function. As described in Section 4.2.3 the network is fed with a set of labeled data which is split and used for training, validation and testing. First, the loss function compares the output of the network with the provided label, the expected output, and calculates a loss measure. The gradient descent algorithm then adjusts the weights in the network in order to minimize this measure by backpropagating the error through the network, calculating the contribution from each node and adjusting their weights accordingly. For a more in-depth explanation of the gradient descent algorithm and backpropagation see [Goodfellow et al., 2016], but in short it calculates an error gradient and makes a small ”step” based on the learning rate in the direction of a local minimum of the loss function.

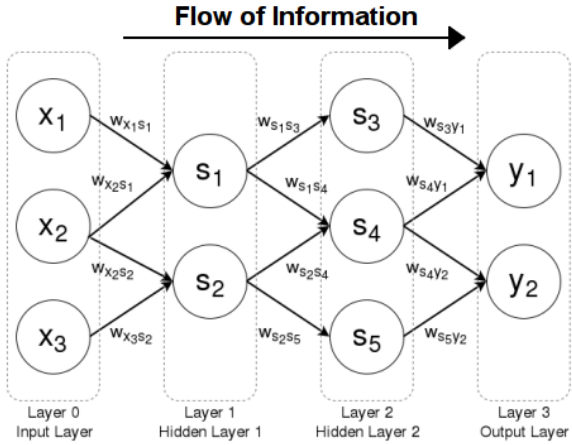


Figure 4.7: Illustration of a feedforward neural network [Patel, 2012].

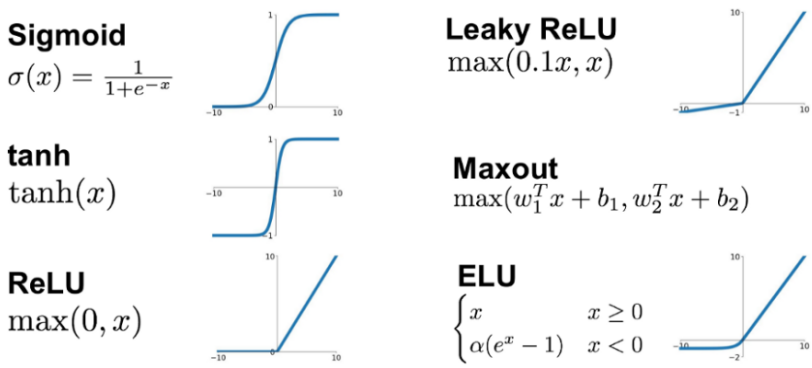


Figure 4.8: Some common activation functions.

4.4.5 Convolutional Neural Network

Convolutional neural network, or CNN, is a deep neural network most commonly used in image-related tasks. This is because of its ability to capture spatial dependencies and being translation invariant. CNN is a type of FFNN, meaning the data only flows in one direction and the layers are fully connected. The network consist of an input layer, hidden layers and an output layer, but the difference lies in how the layers works. The hidden layers typically consist of multiple **convolutional layers**. The most common activation function is the ReLU function, and it is followed by **pooling layers** and fully connected layers. In the convolution process a **filter** (also called a **kernel**) of size MxN is moved around the input with a given step size. At each step a dot product is taken of the filter and an equivalent sized part of the image. The sum of the dot product is then used to make a **feature map**. See Figure 4.9 for an illustration. There can be multiple filters in each convolutional layer resulting in multiple feature maps. The number of feature maps in a layer is called its **depth**. For the input layer the depth is dependent on the type of input. For instance if the input is a RGB image the depth would be three (one for each color), and for a gray scale image the depth would be one. The filter extends through the full depth of its input.

The feature maps from the convolutional layer are then sent to a pooling layer. The pooling layer works in a similar fashion as the convolutional layer, except that instead of the filters taking the dot product, a **pooling operation** is done. The pooling operation is specified rather than learned. The most common pooling operations are; **Max Pooling** - the maximum value is returned, and **Average Pooling** - the average value is returned.

The most common size for a pooling filter is 2x2. This means that after each pooling layer every feature map will have its size reduced by a factor of two. The main purpose of the pooling layer is to reduce the data size in order to ease the computational load and to make the network approximately invariant to local translation. As the pooled feature maps take

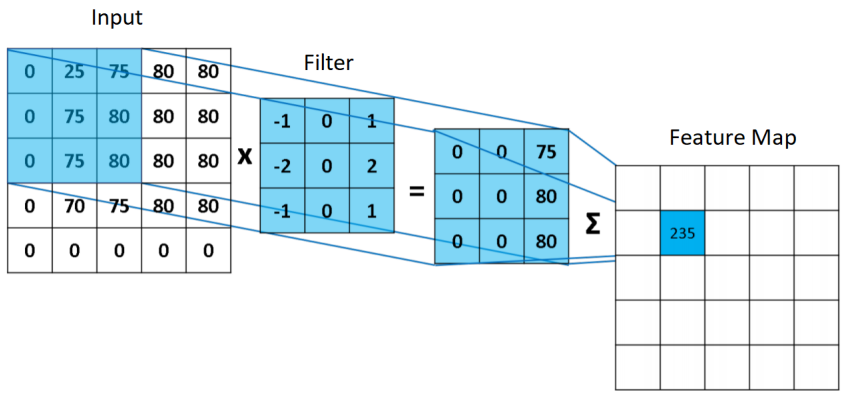


Figure 4.9: Illustration of a filter in a convolutional layer used to create a feature map.

the average or maximum value of the input, small changes in location will not affect the output a lot as it does not look at individual values, but at small areas.

The convolution and pooling process are then repeated. At the end there are one or more fully connected layers. Before the data can be processed by the fully connected layer it has to be flattened as the fully connected layer only can handle 1d data, and not 2d data which is used by the convolutional- and pooling filters. The final fully connected layer is the output layer. See Figure 4.10 for an illustration.

Comparison to Wavelet Scattering

As briefly discussed in Section 2.3.3 the architecture of a CNN and wavelet scattering share many similarities, like that they use filters to create features from the input. The most important difference is that the filters in a CNN have to be learned whereas the filters in wavelet scattering are predefined as wavelets. There are multiple advantages and disadvantages with both methods. The most obvious advantage for using predefined filters is that the only thing that has to be learned are the parameters in the final classifier. This greatly reduces the amount of data needed to get a good performance. As there is a limited amount of data (there are hopefully not too many faults in the power grid each year) this is a very useful feature of wavelet scattering. The downside is that there is a always a possibility that the filters that the CNN learns are better at capturing the traits of the faults compared to wavelets.

4.5 Evaluation Metrics

There are many methods used to evaluate performance of machine learning methods. Here we present the one used in this thesis.

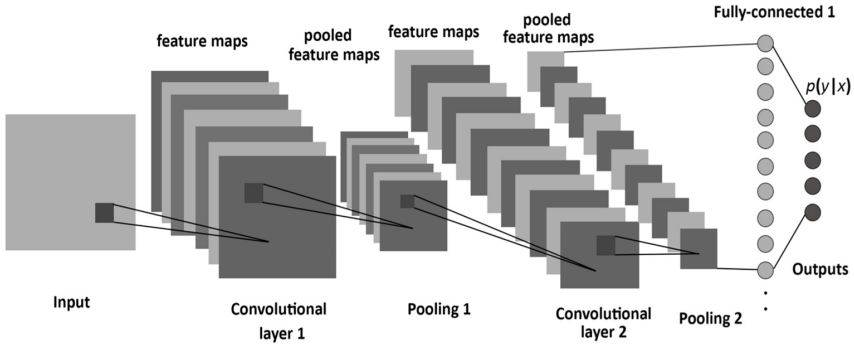


Figure 4.10: Illustration of the architecture of a convolutional neural network [Albelwi and Mahmood, 2017].

4.5.1 Receiver Operating Characteristic Curves

Receiver Operating Characteristic metrics, hence called ROC, can be used to measure classifier output quality of a balanced data-set. 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. The ROC 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) of the Receiver Operating Characteristic Curve (ROC) is calculated to evaluate the classifier. A greater AUC-ROC score is associated with a better classification. A perfect prediction (TPR=1 when FPR=0) is used as a perfect score with an AUC-ROC of 1, and a linear scaling (FPR = TPR) is used as the baseline as a truly random classifier, with an AUC-ROC of 0.5 [Zweig and Campbell, 1993].

Chapter 5

Related Work

In this chapter we will introduce and compare work which is related to the work done in this thesis. There has not been much work regarding prediction of power grid faults, but some on classification. As classification is a much simpler task it is of limited usefulness. It may however provide some insight into useful features and methods that might work with prediction. There has also been very limited work on the usage of wavelet scattering and wavelet transform for similar tasks. There has been some use of wavelet scattering and some using the wavelet transform, but the usage of wavelet transform has mainly been to denoise the signal and not for feature extraction. As we did not find enough resources from research published at well-known conferences, we have also gathered inspiration from Kaggle competitions and some blog posts. Even though such sites do not require published work to undergo a review, Kaggle has an active community that discusses admissions and the blog posts also have many commenters. These sources have been read with some scepticism and mostly been used for inspiration and not as a foundation.

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 using RMS values. There have been written two master's theses in collaboration with the EarlyWarn project [Santi, 2019] and [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 methods, and if so, to what extent. The background related to power grids in this thesis has mostly been based on these two theses in addition to the sources mentioned in them. These theses differ from ours in the way that we want to explore the usability and potential of the raw signal wave as well as compare different feature extraction methods. From the literature search done in [Santi, 2019] it

was concluded that both Fourier transform based methods and wavelet transform showed potential as feature extraction methods. It does however look like only Fourier transform was used in the experiments. As most of the harmonics had a zero value it was decided to only use the 16 first harmonics which seemed to have most significance. It was also discovered that the performance improved when using both the minimum and maximum value of the harmonics rather than just the mean value. For classification methods decision trees, neural networks, SVMs and Bayesian classifiers were said to have the most potential. In the experiments all of these were used except Bayesian Classifiers. Random forest achieved the best performance, followed by neural networks and SVMs. SVMs had the absolute worst performance over all the metrics. For their best performing random forest they achieved 74% accuracy when comparing all the faults versus non-faults, 87% for power interruptions, 70% for ground faults, 63% for voltage dips and 57% for rapid voltage changes. However, later inspection of the data used for these experiments revealed an error in how the DDG was using the overlap period which resulted in some observations that had a long time duration before the fault occurred to have other faults occurring in the same time interval. These faults were a lot easier to predict as they essentially became a classification problem instead of predication. There were also an error with how the non-faults in the data-sets were put together which resulted in duplicates. This is a problem because if one of the duplicates gets put in the training data-set and one in the test data-set, it is a lot easier to predict as the model already has trained on that particular observation. Due to these two problems the scores might have gotten inflated and it is probably not realistic to achieve such high scores.

[Hoffmann et al., 2019] looked into prediction using cycle-by-cycle RMS voltages. The data that was used had 30 minutes per observation resulting in 540,000 samples per observation. In total there were 4101 observations with non-faults, 1940 with voltage sags, 132 with power interruptions, and 1433 with ground faults (observations with missing values were excluded). They chose *gradient boosted decision trees* (similar to CatBoost, Section 4.4.3) as their machine learning model and made one binary classifier for each fault. They used power spectral densities at different frequencies for features. To calculate the spectra they variated over the time interval – from 40 to 1280 seconds –, forecast horizon – from 0 to 40 seconds –, and number of frequency components – from 8 to 64–, totaling in 264 combinations. They achieved 95% prediction rate for power interruptions with a false positive rate of only 20%, AUC score of >0.8 for ground faults and >0.7 for voltage sags. This validates the work in [Santi, 2019] where which found that interruptions are the easiest to predict, followed by ground faults and voltage sags. From the analysis of the results they concluded that it should be possible to increase the forecast horizon beyond 40 seconds, and that more samples equalled better predictive performance. The biggest difference is that we want to focus on using the raw wave signal and not the RMS signal. As they achieved very promising results with gradient boosted decision trees, we think they should be worth looking into further using our data. It might also be interesting to try multi-classification and to classify binary non-faults versus all faults collected in one class.

5.2 Detection of Faults

[Mahela et al., 2015] did a review on detection and classification methods used on power quality events. First they reviewed feature extraction methods, including many of which we think have much potential such as Fourier transform based methods (See Section 2.3.1), S-transform based methods (S-transform is a generalized version of the STFT [See Section 2.3.1] which enables varying window sizes much like the wavelet transform [See Section 2.3.3]) and wavelet transform based methods (Section 2.3.3). They also mentioned other methods we will not explore further in order to limit the scope, and some denoising techniques. They claimed (based on experiments conducted in [Gaouda and Salama, 2009]) that the wavelet transform is better than the STFT. This claim is also backed by [Mallat, 2012]. As such it might be more interesting to focus on wavelet transform based methods instead of STFT for the majority of our experiments. As S-transform is more or less a wavelet transform inspired STFT, we will rather explore just the wavelet transform instead of both to limit the scope and to get more time to experiment with other types of feature extraction methods.

[Mahela et al., 2015] also reviewed classification techniques, including some of which we think have much potential such as SVM based classification (See Section 4.4.1) and neural network based classification (See Section 4.4.4). They also mentioned other methods like Fuzzy systems which we will not go deeper into in order to limit the scope. There are extremely many different variations of NNs, and as they did not specify the architectures they used when doing the review, it is difficult to interpret and make use of their results. They claimed that NN has a better capacity of knowledge representation than SVM, but also claimed that NN is more susceptible to noise.

[Gopakumar et al., 2015] looked into transmission line fault detection using PMU (See Section 3.2) measurements. As PMU sensors only sample at 50Hz and the data we use are sampled with a PQA with a sampling rate greater than 25kHz, the methods they used might not work as well with our data as the difference in frequency is too big. They did also look into identification of location of transmission line fault. As this is out of our scope we will ignore that part and just focus on the detection part. For feature extraction they utilized the Fourier transform on the **EVPA** (Equivalent Voltage Phasor Angle). The EVPA is under normal operating conditions, like the RMS value (See Section 2.2.3), constant. By analysing the frequency domain of this value for deviations, they were able to detect if a fault has occurred. These deviations are caused by harmonic currents which are generated by waveform distortions occurring because of transmission line faults. They did not specify how they did the classification. They only looked at the first 10 harmonics and classified the faults only based on these values. It looks like they classified everything which had values other than zero for any harmonic other than the 1st harmonic were classified as a fault. The fault was then further classified looking at the other harmonics. This model seems to completely ignore noise, and would probably not perform well in our situation. It is not clear whether they used artificially created data or real data in their experiments. However, classifying faults using harmonics looks promising and should be explored further.

5.3 Kaggle and Blog Posts

The competition in [Kaggle, 2019] challenged people to detect power line faults. The data-set that was used came from a real environment, not simulated, and contained a lot of background noise. Each observation contained 800,000 samples taken over 20 milliseconds, giving an extremely high resolution compared to the one we are working with. As the underlying electric grid operated at the same frequency as hours – 50Hz – each observation covered a complete grid cycle. They also had a three phase power system (See Section 2.2.3), as we do. They did however not have access to phase-to-phase measurements, only phase-to-ground. To limit our scope we have decided to only explore phase-to-phase measurements. There were many interesting submissions like the 1st place [mark4h, 2019] which used simple features like RMS values and peak counts in order to classify the faults. Peaks were defined as local maxima which were calculated over different window sizes, see Figure 5.1 for an illustration. To remove the phase a ”flatiron” function was used. In short this function centered all samples around zero, see Figure 5.2 for an illustration.

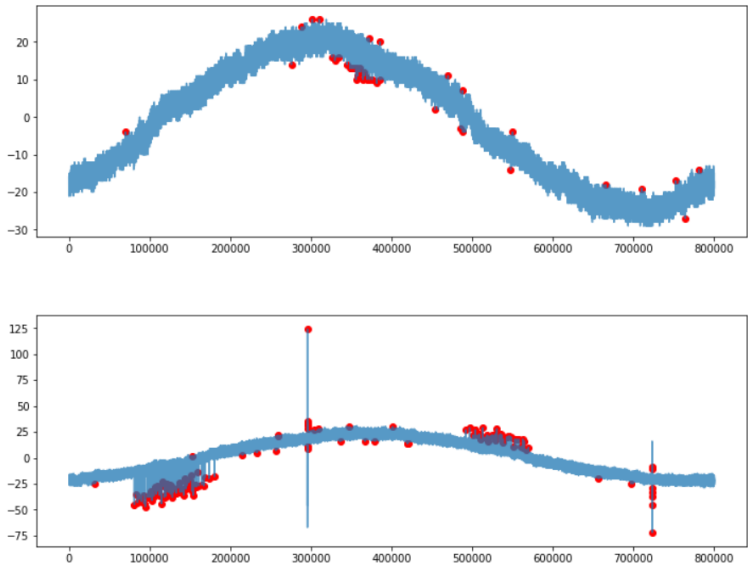


Figure 5.1: Two examples of a wave and its peaks [mark4h, 2019].

Using peaks as a feature looks promising and might be worth looking more into. The flatiron function might also be useful in order to normalize the data. Gradient boosted decision trees gave very good results, which further strengthen our motivation to try them.

In the blog post [Ataspinar, 2018] it was attempted to use wavelet transform spectograms in combination with a CNN (See Section 4.4.5) to classify brain activity signals. The training-set contained 7352 observations where each observation had 128 samples and 9

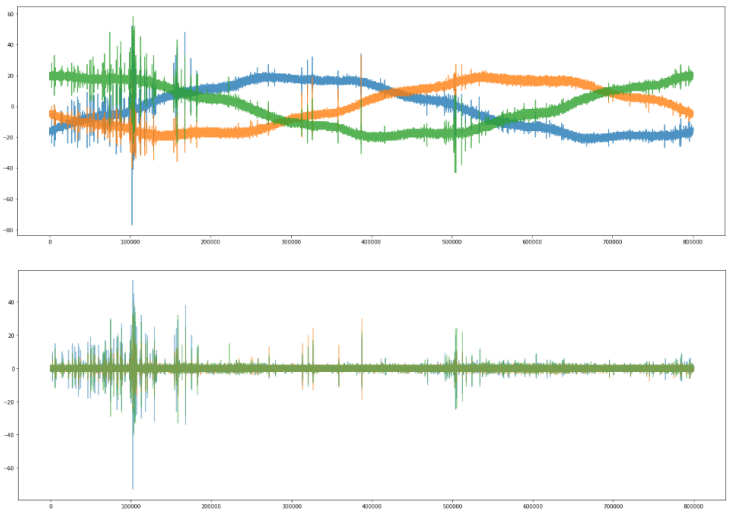


Figure 5.2: An example of a three phase power signal with the phase removed [mark4h, 2019].

components. Compared to the data we use they have a bigger data-set, but less samples per observation. They have less components, 9 compared to our potential 12 (6 phase-to-ground when using max and min for every phase, and 6 phase-to-phase when using max and min for every phase). A spectogram was made from each component resulting in 9 spectograms per observation. To be able to feed this into the CNN they were stacked creating one single image with 9 channels (one channel for every spectogram). They used a very simple CNN with 2 convolutional layers, 2 pooling layers and 2 fully connected layers at the end. The use of wavelet transform spectograms combined with a CNN looks very promising and should be looked into. There might however be a problem regarding the size of the data-set as CNNs need a lot of data in order to achieve good performances.

5.4 Summary

For feature extraction methods Fourier transform based methods and wavelet transform methods seem to have the most potential and should be explored further.

For classification methods [Santi, 2019] achieved good performance with random forest and neural network, [Hoffmann et al., 2019] achieved good performance with gradient boosted decision trees. In addition to these SVMs should be looked at for completeness. CNNs do also seem promising and as it looks like CNNs have not been used a lot in this field of research it might be interesting to test them and see how they perform compared to the others.

Chapter 6

Data

In this chapter we will introduce the data which will be used in this thesis.

6.1 Data

The data-sets used in the experiments in this thesis are presented in Tables 6.1 through 6.26. The data-sets were created using SINTEF's DDG, using fault lists created by their A-HA system, and the labeling scheme explained in Section 7.1.1. As noted in Section 7.1.4, the A-HA system also categorizes "Rapid voltage changes" as may be seen in [Santi, 2019], however, these were not explored in this thesis. To limit the scope we decided to only use phase-to-ground and not phase-to-phase. For wavelet transform based methods we also decided to only use the max value for the first phase.

For binary classification the data-sets used in the experiments were balanced by filtering out random observations from the type with more observations until there was an equal amount of each type. For multi-labeled classification all the data was used.

6.2 Preprocessing

The only thing that had to be addressed concerning the data were missing values. For the raw signal wave data-sets all data that had missing values were removed. For the RMS value- and Fourier coefficient data-sets all data-sets with a missing value rate above 0.01% were removed. The missing values in the remaining RMS value- and Fourier coefficient

Extraction parameter	**Value**
Total Duration	60s
Number of samples	60,000
Time before fault	60s
Resolution	1kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	1000
Ground faults	1000
Power interruptions	108

Table 6.1: Data-set 1. A 1kHz wave form data-set.

Extraction parameter	**Value**
Total Duration	60s
Number of samples	600,000
Time before fault	60s
Resolution	10kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	1000
Ground faults	1000
Power interruptions	108

Table 6.2: Data-set 2. A 10kHz wave form data-set.

Extraction parameter	**Value**
Total Duration	60s
Number of samples	1,500,000
Time before fault	60s
Resolution	25kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	1000
Ground faults	1000
Power interruptions	108

Table 6.3: Data-set 3. A 25kHz wave form data-set.

Extraction parameter	**Value**
Total Duration	60s
Number of samples	3,000,000
Time before fault	60s
Resolution	50kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	1000
Ground faults	1000
Power interruptions	108

Table 6.4: Data-set 4. A 50kHz wave form data-set.

Extraction parameter	**Value**
Total Duration	60s
Number of samples	1,500,000
Time before fault	60s
Resolution	25kHz
Data type	RMS values
Aggregation method	Minimum, Maximum
Overlap period	600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	1000
Ground faults	1000
Power interruptions	108

Table 6.5: Data-set 5. A 25kHz RMS value data-set.

Extraction parameter	**Value**
Total Duration	60s
Number of samples	1,200,000
Time before fault	60s
Resolution	20kHz
Data type	Fourier coefficients
Aggregation method	Minimum, Maximum
Overlap period	600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	1000
Ground faults	1000
Power interruptions	108

Table 6.6: Data-set 6. A 25kHz Fourier coefficient data-set.

Extraction parameter	**Value**
Total Duration	60s (1 second every minute)
Number of samples	60,000
Time before fault	59s
Resolution	1kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	1000
Ground faults	1000
Power interruptions	101

Table 6.7: Data-set 7. A 1kHz wave form data-set.

Extraction parameter	**Value**
Total Duration	60s (1 second every minute)
Number of samples	600,000
Time before fault	59s
Resolution	10kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	1000
Ground faults	1000
Power interruptions	101

Table 6.8: Data-set 8. A 10kHz wave form data-set.

Extraction parameter	**Value**
Total Duration	60s (1 second every minute)
Number of samples	1,500,000
Time before fault	59s
Resolution	25kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	1000
Ground faults	1000
Power interruptions	106

Table 6.9: Data-set 9. A 25kHz wave form data-set.

Extraction parameter	**Value**
Total Duration	60s (1 second every minute)
Number of samples	3,000,000
Time before fault	59s
Resolution	50kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	1000
Ground faults	1000
Power interruptions	101

Table 6.10: Data-set 10. A 50kHz wave form data-set.

Extraction parameter	**Value**
Total Duration	60s (1 second every minute)
Number of samples	1,500,000
Time before fault	59s
Resolution	25kHz
Data type	RMS values
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	1000
Ground faults	1000
Power interruptions	101

Table 6.11: Data-set 11. A 25kHz RMS value data-set.

Extraction parameter	**Value**
Total Duration	60s (1 second every minute)
Number of samples	1,500,000
Time before fault	59s
Resolution	25kHz
Data type	Fourier coefficients
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	990
Ground faults	1000
Power interruptions	101

Table 6.12: Data-set 12. A 25kHz Fourier coefficient data-set.

Extraction parameter	**Value**
Total Duration	3600s
Number of samples	3,600,000
Time before fault	0s
Resolution	1kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	939
Ground faults	1000
Power interruptions	97

Table 6.13: Data-set 13. A 0 minutes before fault 1kHz wave form data-set.

Extraction parameter	**Value**
Total Duration	3540s
Number of samples	3,540,000
Time before fault	60s
Resolution	1kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	939
Ground faults	1000
Power interruptions	97

Table 6.14: Data-set 14. A 1 minute before fault 1kHz wave form data-set.

Extraction parameter	**Value**
Total Duration	3300s
Number of samples	3,300,000
Time before fault	300s
Resolution	1kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	939
Ground faults	1000
Power interruptions	97

Table 6.15: Data-set 15. A 5 minutes before fault 1kHz wave form data-set.

Extraction parameter	**Value**
Total Duration	3000s
Number of samples	3,000,000
Time before fault	600s
Resolution	1kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	939
Ground faults	1000
Power interruptions	97

Table 6.16: Data-set 16. A 10 minutes before fault 1kHz wave form data-set.

Extraction parameter	**Value**
Total Duration	2700s
Number of samples	2,700,000
Time before fault	900s
Resolution	1kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	939
Ground faults	1000
Power interruptions	97

Table 6.17: Data-set 17. A 15 minutes before fault 1kHz wave form data-set.

Extraction parameter	**Value**
Total Duration	1800s
Number of samples	1,800,000
Time before fault	1800s
Resolution	1kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	939
Ground faults	1000
Power interruptions	97

Table 6.18: Data-set 18. A 30 minutes before fault 1kHz wave form data-set.

Extraction parameter	**Value**
Total duration	600s
Number of samples	600,000
Time before fault	3000s
Resolution	1kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	939
Ground faults	1000
Power interruptions	97

Table 6.19: Data-set 19. A 50 minutes before fault 1kHz wave form data-set.

Extraction parameter	**Value**
Total duration	600s
Number of samples	600,000
Time before fault	0s
Resolution	1kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	939
Ground faults	1000
Power interruptions	97

Table 6.20: Data-set 20. A 0 minutes before fault 1kHz wave form data-set.

Extraction parameter	**Value**
Total duration	600s
Number of samples	600,000
Time before fault	60s
Resolution	1kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	939
Ground faults	1000
Power interruptions	97

Table 6.21: Data-set 21. A 1 minute before fault 1kHz wave form data-set.

Extraction parameter	**Value**
Total duration	600s
Number of samples	600,000
Time before fault	300s
Resolution	1kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	939
Ground faults	1000
Power interruptions	97

Table 6.22: Data-set 22. A 5 minutes before fault 1kHz wave form data-set.

Extraction parameter	**Value**
Total duration	600s
Number of samples	600,000
Time before fault	600s
Resolution	1kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	939
Ground faults	1000
Power interruptions	97

Table 6.23: Data-set 23. A 10 minutes before fault 1kHz wave form data-set.

Extraction parameter	**Value**
Total duration	600s
Number of samples	600,000
Time before fault	900s
Resolution	1kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	939
Ground faults	1000
Power interruptions	97

Table 6.24: Data-set 24. A 15 minutes before fault 1kHz wave form data-set.

Extraction parameter	**Value**
Total duration	600s
Number of samples	600,000
Time before fault	1800s
Resolution	1kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	939
Ground faults	1000
Power interruptions	97

Table 6.25: Data-set 25. A 30 minutes before fault 1kHz wave form data-set.

Extraction parameter	**Value**
Total duration	600s
Number of samples	600,000
Time before fault	3000s
Resolution	1kHz
Data type	Wave form
Aggregation method	Minimum, Maximum
Overlap period	3600s
Specificity	V1, V2, V3
Fault types	**Successfully extracted**
Non-faults	2100
Voltage sags	939
Ground faults	1000
Power interruptions	97

Table 6.26: Data-set 26. A 50 minutes before fault 1kHz wave form data-set.

data-sets were interpolated using linear interpolation. The pruning of the RMS value- and Fourier coefficient data-sets was less strict because the data-sets would be too small if a more strict pruning was done.

6.3 Feature Extraction

The feature extraction methods used in the experiments in this thesis are presented here. The wavelet transform was implemented using the pywavelets library [Lee et al., 2019], the spectograms were made using matplotlib [Hunter, 2007] and PILLOW [Lundh and Clark, 1995], the scattering was implemented using the kymatio library [Andreux et al., 2018].

6.3.1 Wavelet Transform Spectograms

In this method the signal was first transformed by a continuous wavelet transform using 2 sets of scales. The first set of scales had every scale ranging from 1 to 32. This was chosen as the mother wavelet with these scales covered frequencies from 812Hz at scale 1, to 25Hz at scale 32, and that the most interesting noise should be in that frequency range. The second set had every 16th scale ranging from 1 to 3750. This was chosen as the signal had 60,000 samples and the mother wavelet had a length of 16, meaning at the largest scale the wavelet would cover the whole signal (60,000/16 = 3750). Using every 16th scale, and not every scale was decided as we did not have the memory capacity, as well as the processing time got unreasonably long. Different types of mother wavelets were used:

- Morlet
- Mexican hat
- Haar

which were presented in Figure 2.9. The coefficients gotten from the transform were then used to create a spectogram. This was done both as a grayscale spectogram and also using various colormaps. As there were more samples than there were scales (60,000 versus 235 – $\frac{3750}{16} \approx 235$ –, or 32) we tried rectangular images with a bigger width than height, but also square images which seemed to be the norm. The output coefficients given from the continuous wavelet transform are represented by a matrix with shape $M \times N$ where M is the number of scales and N is the number of samples. We tried using different image sizes with heights up to 235 and widths up to 4 times the height, limited by memory capacity. As the spectogram image was of a smaller size than the coefficient matrix, it was down-sampled using cubic spline interpolation.

6.3.2 Wavelet Scattering

In this method the signal was decomposed using wavelet scattering, using a combination of parameters:

- **J**: 6, 8, 12, 15
- **Q**: 12, 24, 48, 96
- **T**: 60,000, 1,500,000

where J is the maximum log-scale, meaning the maximum scale is given by 2^J, Q is the number of first-order wavelets per octave, and T is the length of the signal. As most of the structures we are interested in are in the first and second order, we remove the zeroth order coefficients. To increase discriminability, we took the logarithm of the coefficients gotten from the scattering. Before taking the logarithm a small constant of 10^{-15} was added in order to prevent values close to zero from getting extremely dominant. Finally we averaged along the time dimension to make it invariant to time-shift.

6.3.3 Aggregated Values

In this method the values were aggregated to represent the different extracted features. The aggregated values which were calculated were the minimum (min), maximum (max), mean, standard deviation (STD), and the signal-to-noise ratio (SNR) of the normalized signal. SNR was suggested as a possible aggregated value in [Jahr and Meen, 2019], and is calculated as $\frac{\text{mean}}{\text{STDS}}$. The signal was normalized by dividing the values given by DDG by the voltage of the power line.

Combined Values

In this version the values were calculated for all phases – V1, V2, and V3 – for the values given by both the aggregation method in the data-sets – max and min – and additionally calculated for the difference, max-min, resulting in a total of 45 values for one time-series.

Singular Values

In this version the values were calculated for all phases – V1, V2, and V3 – for the values given by both the aggregation method in the data-sets – max and min – and additionally calculated for the difference, max-min, for each second. Additionally, the difference between the same aggregated value was calculated for each second, resulting in a total of $45 \times 60 + 45 \times (60 - 1) = 5355$ values for one time-series.

6.3.4 Fourier

In this method the signal was decomposed using the Fourier transform. Only the 16 first harmonics were used, the rest were discarded. The 16 first harmonics were chosen based on the reasoning from Section 5.1.

Chapter 7

Exploration

In this chapter we present an Exploratory Data Analysis (EDA) on the reported faults and on our data presented in Chapter 6.

7.1 Fault Distributions

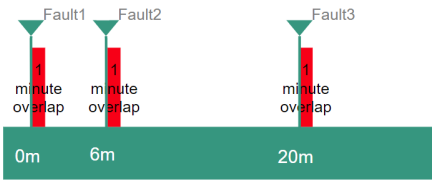
We look closer at the distribution of faults in time and space to see if there is anything major our – or other's – project ought to take into consideration.

7.1.1 Fault Overlapping

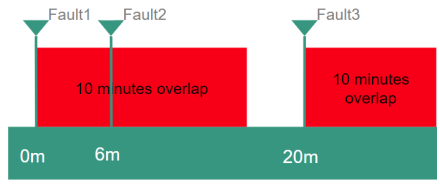
We examine how frequently faults are reported to happen within the overlap period of other faults. We define overlap period as:

> The **overlap period** is the time-frame after a reported fault occurs where any newly reported fault will be considered to be the same fault. If a fault occurs within the overlap period of the previous fault, the overlap window is extended as if it started at the newly reported fault. An example is given in Figure 7.1.

The results are shown in Table 7.1. It is immediately apparent that many of the reported faults overlap to some degree. Even when looking at an overlap period of 0.1 seconds, over half of the faults are considered overlapped. Increasing the overlap window increases this number significantly, ground faults are the most clear example of this. While most



(a) 1 minute overlap. No fault occurs within another faults overlap period, all three faults are considered separate faults.



(b) 10 minute overlap. Fault2 occurs within the overlap period of Fault1, ignoring it and extending the overlap period of Fault1. Fault1 and Fault3 are considered separate faults.

Figure 7.1: 3 reported faults occurring at 0, 6, and 20 minutes, with different overlap periods.

	Faults retained with overlap period							
Fault type	**0.1 seconds**		**1 second**		**10 seconds**		**1 minute**	
GF	8197	(53.27%)	4210	(41.82%)	2626	(36.52%)	2080	(35.25%)
VS	5104	(33.17%)	4033	(40.06%)	3124	(43.44%)	2673	(45.30%)
GF/VS	1684	(10.94%)	1438	(14.28%)	1056	(14.69%)	775	(13.13%)
PI	176	(1.14%)	141	(1.40%)	133	(1.85%)	127	(2.15%)
RVC	120	(0.78%)	120	(1.19%)	119	(1.65%)	112	(1.90%)
RVC/VS	56	(0.36%)	56	(0.56%)	57	(0.79%)	53	(0.90%)
GF/PI	22	(0.14%)	33	(0.33%)	34	(0.47%)	31	(0.53%)
PI/VS	11	(0.07%)	15	(0.15%)	19	(0.26%)	20	(0.34%)
GF/RVC	9	(0.06%)	9	(0.09%)	9	(0.13%)	12	(0.20%)
GF/PI/VS	6	(0.04%)	10	(0.10%)	11	(0.15%)	11	(0.19%)
GF/RVC/VS	3	(0.02%)	3	(0.03%)	3	(0.04%)	6	(0.10%)
PI/RVC/VS	0	(0.00%)	0	(0.00%)	0	(0.00%)	1	(0.02%)
Total	15388		10068		7191		5901	

Table 7.1: The amount of each separate fault, and the percentage of total separate faults, using different overlap periods when using our labeling scheme. There were 34433 reported faults, where 26425 were ground faults, 7445 were voltage sags, 285 were power interruptions, 188 were rapid voltage changes, and 90 were discarded due to containing errors. Merged faults are listed as both, separated with a forward slash.
The abbreviations for the faults are: Voltage sag: VS, Ground fault: GF, Power interruption: PI, Rapid voltage change: RVC.

	Faults retained with overlap period							
Fault type	**0.1 seconds**		**1 second**		**10 seconds**		**1 minute**	
GF	9721	(63.17%)	5520	(54.83%)	3553	(49.41%)	2727	(46.21%)
VS	5302	(34.46%)	4218	(41.90%)	3316	(46.11%)	2870	(48.64%)
RVC	186	(1.21%)	186	(1.85%)	185	(2.57%)	174	(2.95%)
PI	179	(1.16%)	144	(1.43%)	137	(1.91%)	130	(2.20%)
Total	15388		10068		7191		5901	

Table 7.2: The amount of each separate fault, and the percentage of total separate faults, using different overlap periods when using the DDG labeling scheme. There were 34433 reported faults, where 26425 were ground faults, 7445 were voltage sags, 285 were power interruptions, 188 were rapid voltage changes, and 90 were discarded due to containing errors. Merged faults are listed as both, separated with a forward slash.
The abbreviations for the faults are: Voltage sag: VS, Ground fault: GF, Power interruption: PI, Rapid voltage change: RVC.

faults only overlap faults of the same type as themselves, or overlap nothing at all, a not insignificant amount of the faults are overlapping other types of faults. When looking at an overlap period of 1 minute, 15% of the separate faults are considered to be a combination of faults. We took note that DDG will label any series of faults as the initial fault of that overlap period. This means that for instance, if a series of faults is reported as a ground fault and later as a power interruption, it will by DDG be labeled as a ground fault. We note that this might cause problems for machine learning methods. For instance, if we look at the wave leading up to a power interruption, it might carry the characteristics of this fault type in it. However, if A-HA label the early onset signs of the power interruption as i.e. a ground fault, our classifiers will incorrectly learn that this wave carried the characteristics of a ground fault.

When using the DDG labeling scheme, Table 7.1 is rather displayed as Table 7.2, where amongst other problems, almost all faults that were a combination of ground faults and voltage sags are now labeled as ground faults. As noted earlier are a significant amount of the separate faults a combinations of faults, and as such, we do suggest that a more strict labeling scheme ought to be used in DDG if fault types are to be compared to each other. For example, giving degrees of importance to different fault types, and labeling the fault as the fault type it contains with the highest importance, such as giving the aforementioned example a power interruption label rather than a ground fault label, due to power interruptions being a more severe type of fault than ground faults. Another method could be to apply some kind of fuzzy labeling, where the combined fault would belong partially to each of the fault types it contains.

7.1.2 Faults Leading Into Other Faults

Next we examine how often separate faults occur shortly after each other. We consider faults to happen shortly after each other if they occur outside of each others overlap pe-

	Sequences found with overlap period							
Sequences	**0.1 seconds**		**1 second**		**10 seconds**		**1 minute**	
GF/VS → GF	99	(1.21%)	67	(1.59%)	58	(2.21%)	44	(2.12%)
GF → GF/VS	91	(5.40%)	64	(4.45%)	48	(4.55%)	37	(4.77%)
GF → VS	85	(1.67%)	55	(1.36%)	46	(1.47%)	17	(0.64%)
VS → GF	78	(0.95%)	61	(1.45%)	46	(1.75%)	23	(1.11%)
GF/VS → VS	38	(0.74%)	37	(0.92%)	36	(1.15%)	36	(1.35%)
VS → GF/VS	32	(1.90%)	16	(1.11%)	16	(1.52%)	21	(2.71%)
GF → PI	18	(10.23%)	4	(2.84%)	1	(0.75%)	0	(0.00%)
VS → PI	12	(6.82%)	7	(4.96%)	5	(3.76%)	3	(2.36%)
PI → VS	9	(0.18%)	6	(0.15%)	3	(0.10%)	3	(0.11%)
GF → GF/PI	7	(31.82%)	7	(21.21%)	5	(14.71%)	2	(6.45%)
GF → RVC	7	(5.83%)	7	(5.83%)	7	(5.88%)	3	(2.68%)
GF/PI → GF	5	(0.06%)	3	(0.07%)	3	(0.11%)	3	(0.14%)
VS → RVC	4	(3.33%)	4	(3.33%)	3	(2.52%)	2	(1.79%)
GF/VS → PI	3	(1.70%)	0	(0.00%)	0	(0.00%)	0	(0.00%)
PI → GF	3	(0.04%)	1	(0.02%)	1	(0.04%)	1	(0.05%)
RVC/VS → GF	3	(0.04%)	3	(0.07%)	3	(0.11%)	1	(0.05%)
VS → RVC/VS	3	(5.36%)	3	(5.36%)	2	(3.51%)	2	(3.77%)
RVC/VS → VS	3	(0.06%)	3	(0.07%)	4	(0.13%)	2	(0.07%)
GF/VS → GF/PI/VS	2	(33.33%)	0	(0.00%)	0	(0.00%)	0	(0.00%)
GF/PI/VS → GF	2	(0.02%)	2	(0.05%)	0	(0.00%)	0	(0.00%)
GF/RVC → GF	2	(0.02%)	2	(0.05%)	2	(0.08%)	1	(0.05%)
GF → GF/RVC	2	(22.22%)	2	(22.22%)	2	(22.22%)	1	(8.33%)
RVC/VS → RVC	2	(1.67%)	2	(1.67%)	2	(1.68%)	2	(1.79%)
GF/RVC/VS → GF	2	(0.02%)	2	(0.05%)	1	(0.04%)	2	(0.10%)
PI/VS → VS	2	(0.04%)	2	(0.05%)	2	(0.06%)	1	(0.04%)
VS → GF/PI/VS	1	(16.67%)	1	(10.00%)	2	(18.18%)	1	(9.09%)
PI/VS → PI	1	(0.57%)	2	(1.42%)	2	(1.50%)	0	(0.00%)
GF/PI/VS → VS	1	(0.02%)	1	(0.02%)	1	(0.03%)	2	(0.07%)
GF → GF/PI/VS	1	(16.67%)	2	(20.00%)	1	(9.09%)	0	(0.00%)
GF/PI → VS	1	(0.02%)	2	(0.05%)	2	(0.06%)	1	(0.04%)

Table 7.3: The frequencies of faults occurring within **5 minutes** of faults of another type occurring, with different overlap periods. Faults merged due to overlap period are listed as both, separated with a forward slash. The percentage of the resulting faults that are caused by this sequence of faults are listed. Only sequences which had more than one occurrence for at least one overlap period are listed.
The abbreviations for the faults are: Voltage sag: VS, Ground fault: GF, Power interruption: PI, Rapid voltage change: RVC.

	Sequences found with overlap period			
Sequences	**0.1 seconds**	**1 second**	**10 seconds**	**1 minute**
GF/VS → GF	110 (1.34%)	79 (1.88%)	71 (2.70%)	60 (2.88%)
GF → GF/VS	102 (6.06%)	76 (5.29%)	61 (5.78%)	53 (6.84%)
GF → VS	100 (1.96%)	66 (1.64%)	56 (1.79%)	24 (0.90%)
VS → GF	95 (1.16%)	77 (1.83%)	62 (2.36%)	34 (1.63%)
GF/VS → VS	52 (1.02%)	53 (1.31%)	53 (1.70%)	54 (2.02%)
VS → GF/VS	41 (2.43%)	26 (1.81%)	26 (2.46%)	33 (4.26%)
GF → PI	19 (10.80%)	5 (3.55%)	2 (1.50%)	1 (0.79%)
VS → PI	13 (7.39%)	8 (5.67%)	6 (4.51%)	4 (3.15%)
PI → VS	11 (0.22%)	8 (0.20%)	4 (0.13%)	4 (0.15%)
GF → GF/PI	7 (31.82%)	7 (21.21%)	6 (17.65%)	3 (9.68%)
GF → RVC	7 (5.83%)	7 (5.83%)	7 (5.88%)	3 (2.68%)
VS → RVC	7 (5.83%)	7 (5.83%)	6 (5.04%)	5 (4.46%)
GF/PI → GF	5 (0.06%)	3 (0.07%)	3 (0.11%)	3 (0.14%)
RVC/VS → VS	4 (0.08%)	4 (0.10%)	5 (0.16%)	3 (0.11%)
GF/VS → PI	3 (1.70%)	0 (0.00%)	0 (0.00%)	0 (0.00%)
PI → GF	3 (0.04%)	1 (0.02%)	1 (0.04%)	1 (0.05%)
RVC/VS → RVC	3 (2.50%)	3 (2.50%)	3 (2.52%)	3 (2.68%)
RVC → RVC/VS	3 (5.36%)	3 (5.36%)	3 (5.26%)	3 (5.66%)
RVC/VS → GF	3 (0.04%)	3 (0.07%)	3 (0.11%)	1 (0.05%)
VS → RVC/VS	3 (5.36%)	3 (5.36%)	2 (3.51%)	2 (3.77%)
RVC → VS	3 (0.06%)	3 (0.07%)	2 (0.06%)	2 (0.07%)
GF/VS → GF/PI/VS	2 (33.33%)	0 (0.00%)	0 (0.00%)	0 (0.00%)
PI/VS → PI	2 (1.14%)	3 (2.13%)	3 (2.26%)	1 (0.79%)
VS → GF/PI/VS	2 (33.33%)	2 (20.00%)	3 (27.27%)	3 (27.27%)
GF/PI/VS → GF	2 (0.02%)	2 (0.05%)	0 (0.00%)	0 (0.00%)
GF/RVC → GF	2 (0.02%)	2 (0.05%)	2 (0.08%)	1 (0.05%)
GF → GF/RVC	2 (22.22%)	2 (22.22%)	2 (22.22%)	2 (16.67%)
GF → RVC/VS	2 (3.57%)	2 (3.57%)	2 (3.51%)	1 (1.89%)
GF/RVC/VS → GF	2 (0.02%)	2 (0.05%)	1 (0.04%)	2 (0.10%)
PI/VS → VS	2 (0.04%)	2 (0.05%)	2 (0.06%)	2 (0.07%)
GF/PI/VS → GF/VS	1 (0.06%)	2 (0.14%)	2 (0.19%)	2 (0.26%)
GF/PI/VS → VS	1 (0.02%)	1 (0.02%)	2 (0.06%)	3 (0.11%)
GF → GF/PI/VS	1 (16.67%)	2 (20.00%)	1 (9.09%)	0 (0.00%)
GF/PI → VS	1 (0.02%)	2 (0.05%)	2 (0.06%)	1 (0.04%)

Table 7.4: The frequencies of faults occurring within **15 minutes** of faults of another type occurring, with different overlap periods. Faults merged due to overlap period are listed as both, separated with a forward slash. The percentage of the resulting faults that are caused by this sequence of faults are listed. Only sequences which had more than one occurrence for at least one overlap period are listed.
The abbreviations for the faults are: Voltage sag: VS, Ground fault: GF, Power interruption: PI, Rapid voltage change: RVC.

	Sequences found with overlap period							
Sequences	**0.1 seconds**		**1 second**		**10 seconds**		**1 minute**	
GF → VS	134	(2.63%)	99	(2.45%)	89	(2.85%)	54	(2.02%)
VS → GF	129	(1.57%)	108	(2.57%)	93	(3.54%)	60	(2.88%)
GF/VS → GF	122	(1.49%)	93	(2.21%)	83	(3.16%)	78	(3.75%)
GF → GF/VS	114	(6.77%)	88	(6.12%)	73	(6.91%)	67	(8.65%)
GF/VS → VS	65	(1.27%)	69	(1.71%)	70	(2.24%)	72	(2.69%)
VS → GF/VS	61	(3.62%)	48	(3.34%)	47	(4.45%)	52	(6.71%)
GF → PI	21	(11.93%)	7	(4.96%)	4	(3.01%)	3	(2.36%)
VS → PI	16	(9.09%)	11	(7.80%)	9	(6.77%)	6	(4.72%)
PI → VS	14	(0.27%)	11	(0.27%)	7	(0.22%)	7	(0.26%)
GF → RVC	9	(7.50%)	9	(7.50%)	9	(7.56%)	5	(4.46%)
VS → RVC	8	(6.67%)	8	(6.67%)	7	(5.88%)	6	(5.36%)
GF → GF/PI	7	(31.82%)	7	(21.21%)	7	(20.59%)	4	(12.90%)
RVC/VS → RVC	7	(5.83%)	7	(5.83%)	7	(5.88%)	7	(6.25%)
VS → RVC/VS	7	(12.50%)	7	(12.50%)	6	(10.53%)	5	(9.43%)
GF/RVC → GF	6	(0.07%)	6	(0.14%)	6	(0.23%)	5	(0.24%)
RVC → VS	6	(0.12%)	6	(0.15%)	5	(0.16%)	5	(0.19%)
PI → GF	5	(0.06%)	3	(0.07%)	3	(0.11%)	2	(0.10%)
GF/PI → GF	5	(0.06%)	3	(0.07%)	3	(0.11%)	4	(0.19%)
RVC/VS → VS	5	(0.10%)	5	(0.12%)	6	(0.19%)	4	(0.15%)
RVC → RVC/VS	4	(7.14%)	4	(7.14%)	4	(7.02%)	4	(7.55%)
RVC/VS → GF	4	(0.05%)	4	(0.10%)	4	(0.15%)	2	(0.10%)
GF/VS → PI	3	(1.70%)	0	(0.00%)	0	(0.00%)	1	(0.79%)
GF → GF/RVC	3	(33.33%)	3	(33.33%)	3	(33.33%)	4	(33.33%)
GF/PI → VS	2	(0.04%)	3	(0.07%)	3	(0.10%)	2	(0.07%)
GF/PI/VS → GF/VS	2	(0.12%)	3	(0.21%)	3	(0.28%)	3	(0.39%)
GF/VS → GF/PI/VS	2	(33.33%)	2	(20.00%)	2	(18.18%)	2	(18.18%)
PI/VS → PI	2	(1.14%)	3	(2.13%)	3	(2.26%)	1	(0.79%)
VS → GF/PI/VS	2	(33.33%)	2	(20.00%)	3	(27.27%)	3	(27.27%)
GF/PI/VS → GF	2	(0.02%)	2	(0.05%)	0	(0.00%)	0	(0.00%)
RVC → GF	2	(0.02%)	2	(0.05%)	2	(0.08%)	2	(0.10%)
GF → RVC/VS	2	(3.57%)	2	(3.57%)	2	(3.51%)	1	(1.89%)
GF/RVC/VS → GF	2	(0.02%)	2	(0.05%)	1	(0.04%)	3	(0.14%)
VS → PI/VS	2	(18.18%)	2	(13.33%)	2	(10.53%)	2	(10.00%)
PI/VS → VS	2	(0.04%)	2	(0.05%)	3	(0.10%)	3	(0.11%)
GF/PI/VS → VS	1	(0.02%)	1	(0.02%)	2	(0.06%)	3	(0.11%)
GF → GF/PI/VS	1	(16.67%)	2	(20.00%)	1	(9.09%)	0	(0.00%)

Table 7.5: The frequencies of faults occurring within **1 hour** of faults of another type occurring, with different overlap periods. Faults merged due to overlap period are listed as both, separated with a forward slash. The percentage of the resulting faults that are caused by this sequence of faults are listed. Only sequences which had more than one occurrence for at least one overlap period are listed.
The abbreviations for the faults are: Voltage sag: VS, Ground fault: GF, Power interruption: PI, Rapid voltage change: RVC.

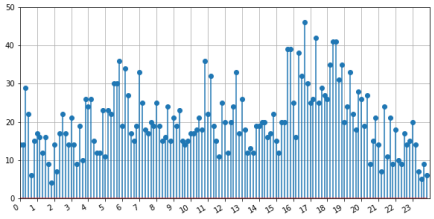
riods, but the latter begins within X minutes of the prior ending. The results are shown in Tables 7.3, 7.4, and 7.5, for $X = 5$, $X = 15$, and $X = 60$ respectively. Around 5% of all ground faults and 5% of voltage sags occur after the converse fault. Looking at how often ground faults happen before power interruptions with different overlap periods signifies the impact overlap duration has on what constitutes separate faults and what constitutes merged faults. It also signifies how often other faults lead into power interruptions, highlighting further the labeling problem of DDG discussed prior.

7.1.3 Time Distribution of Faults

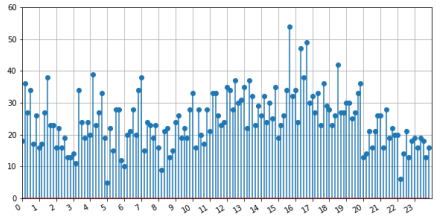
We inspect at what times different faults occur. Some results are presented in Figures 7.2 and 7.3. It appears that ground faults are more likely to happen during summer, while voltage sags appear to happen slightly more frequently around December. Power interruptions appear to happen slightly more during morning hours. We do not conclude in this thesis what the causes of these time distributions are, if they represent an actual tendency of the faults, if they stem from bias in A-HA, or occur due to a lack of sufficient observations. If a tendency actually does exist, including features representing the time of day and/or year might prove beneficial to EarlyWarn.

7.1.4 Fault Distribution for Different Nodes

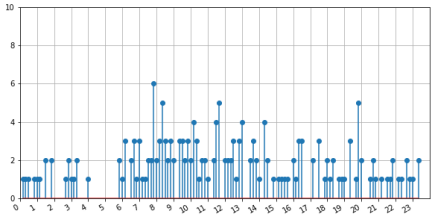
Our data is retrieved from 12 different nodes in the Norwegian power grid, we look at what rates different faults are reported by the different nodes to see if we can see any patterns or outliers. The findings are presented in Tables 7.6 and 7.7. It is apparent from looking at the tables that the faults are **not** evenly distributed. For instance, Node2 is responsible for 45% of all reported power interruptions, while Node3 reports 99% of all rapid voltage changes. Some nodes, such as Node10 and Node11, report almost only voltage sags, while other, such as Node1 and Node6, have equally many or more ground faults than voltage sags. Nodes also report significantly different amount of faults. Node9, Node10, and Node11 each contribute less than 1% of the total number of faults, while Node0 contributes almost 19% of all faults. It is not completely clear whether these differences are the result of natural causes, or if it is caused by a bias or fault in the reporting system. Taking into consideration the statistics presented in Figure 2.16 and Figure 2.18, we can see that most of the operational faults are caused by surroundings (thunderstorms, vegetation, wind) and equipment. As some locations are more vulnerable to some forms of weather and that faulty equipment might be more prone to have reoccurring faults, we think it is fair to assume that the imbalance in reported faults are mostly due to natural causes and not only, if at all, due to bias in A-HA. No matter the cause, the nodes report noticeably different fault rates, and as such it might be reasonable to try to predict faults for each node individually, rather than combined. When mixing faults from all nodes, characteristics associated with e.g. 300kV nets might become correlated with voltage sags, a correlation which the model would have no use for on the other nets, and might even negatively affect



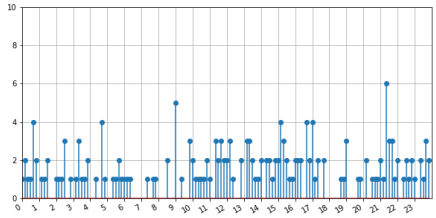
(a) Ground faults



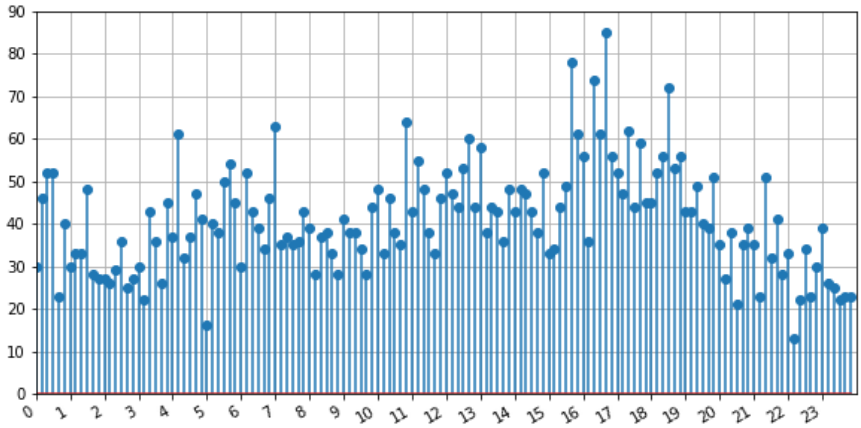
(b) Voltage sags



(c) Power interruptions

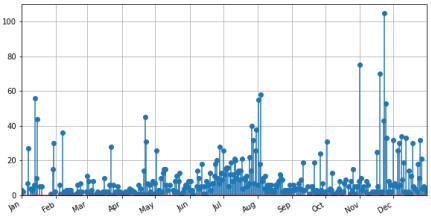


(d) Rapid voltage changes

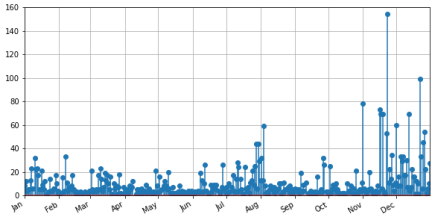


(e) Combined

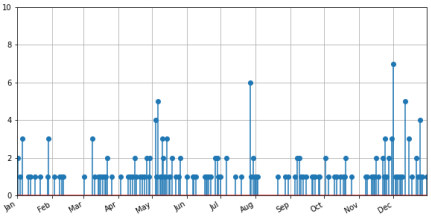
Figure 7.2: Hourly distribution of faults for all merged faults with overlap period of **1 minute**. Faults that were combinations of faults are included in all relevant plots. I.e. a ground fault / voltage sag fault is included in both (a) and (b).



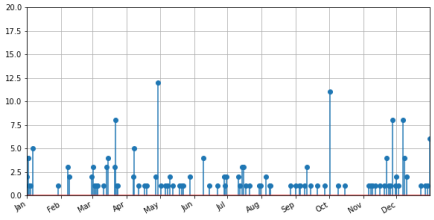
(a) Ground faults



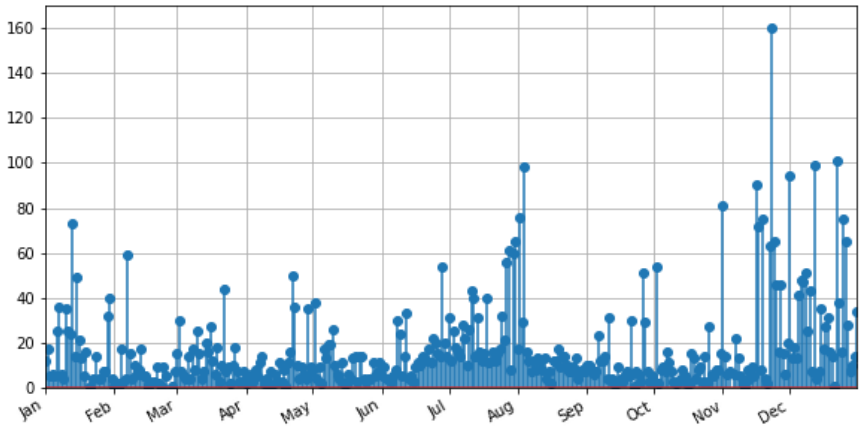
(b) Voltage sags



(c) Power interruptions



(d) Rapid voltage changes



(e) Combined

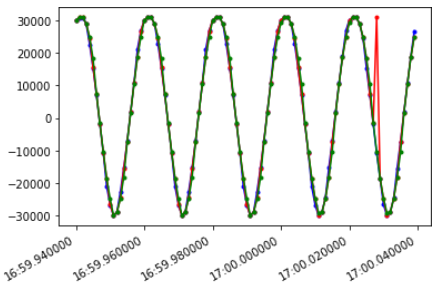
Figure 7.3: Monthly distribution of faults for all merged faults with overlap period of **1 minute**. Faults that were combinations of faults are included in all relevant plots. I.e. a ground fault / voltage sag fault is included in both (a) and (b).

	Instances	Local%	Global%	Global% of this fault
Node0 15kV				
Ground faults	467	36.26%	6.84%	16.02%
Power interruptions	3	0.23%	0.04%	1.58%
Rapid voltage changes	0	0.00%	0.00%	0.00%
Voltage sags	818	63.51%	11.98%	23.11%
Node1 18kV				
Ground faults	427	58.02%	6.25%	14.65%
Power interruptions	18	2.45%	0.26%	9.47%
Rapid voltage changes	0	0.00%	0.00%	0.00%
Voltage sags	291	39.54%	4.26%	8.22%
Node2 22kV				
Ground faults	282	38.42%	4.13%	9.67%
Power interruptions	87	11.85%	1.27%	45.79%
Rapid voltage changes	0	0.00%	0.00%	0.00%
Voltage sags	365	49.73%	5.35%	10.31%
Node3 22kV				
Ground faults	310	30.27%	4.54%	10.63%
Power interruptions	7	0.68%	0.10%	3.68%
Rapid voltage changes	182	17.77%	2.67%	98.91%
Voltage sags	525	51.27%	7.69%	14.83%
Node4 22kV				
Ground faults	304	55.07%	4.45%	10.43%
Power interruptions	5	0.91%	0.07%	2.63%
Rapid voltage changes	0	0.00%	0.00%	0.00%
Voltage sags	243	44.02%	3.56%	6.87%
Node5 22kV				
Ground faults	177	43.60%	2.59%	6.07%
Power interruptions	18	4.43%	0.26%	9.47%
Rapid voltage changes	0	0.00%	0.00%	0.00%
Voltage sags	211	51.97%	3.09%	5.96%

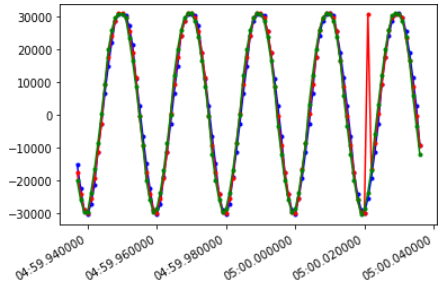
Table 7.6: Fault distribution of faults for all merged faults for nodes with overlap period of **1 minute**. *Local%* indicates how many of that node's fault is of this fault type, *Global%* indicates how many of all the node's faults that are this node and fault type, and *Global% of this fault* indicates how many of this fault type occurred at this node. Faults that were combinations of faults are included in all relevant lines. I.e. a ground fault / voltage sag fault is counted both as a voltage sad and a ground fault for that node. 1/2

	Instances	Local%	Global%	Global% of this fault
Node6 66kV				
Ground faults	536	54.47%	7.85%	18.39%
Power interruptions	19	1.93%	0.28%	10.00%
Rapid voltage changes	0	0.00%	0.00%	0.00%
Voltage sags	429	43.60%	6.28%	12.12%
Node7 66kV				
Ground faults	306	43.71%	4.48%	10.50%
Power interruptions	6	0.86%	0.09%	3.16%
Rapid voltage changes	0	0.00%	0.00%	0.00%
Voltage sags	388	55.43%	5.68%	10.96%
Node8 132kV				
Ground faults	93	39.08%	1.36%	3.19%
Power interruptions	24	10.08%	0.35%	12.63%
Rapid voltage changes	0	0.00%	0.00%	0.00%
Voltage sags	121	50.84%	1.77%	3.42%
Node9 300kV				
Ground faults	13	22.81%	0.19%	0.45%
Power interruptions	1	1.75%	0.01%	0.53%
Rapid voltage changes	0	0.00%	0.00%	0.00%
Voltage sags	43	75.44%	0.63%	1.22%
Node10 300kV				
Ground faults	0	0.00%	0.00%	0.00%
Power interruptions	1	1.49%	0.01%	0.53%
Rapid voltage changes	0	0.00%	0.00%	0.00%
Voltage sags	66	98.51%	0.97%	1.86%
Node11 300kV				
Ground faults	0	0.00%	0.00%	0.00%
Power interruptions	1	2.38%	0.01%	0.53%
Rapid voltage changes	2	4.76%	0.03%	1.09%
Voltage sags	39	92.86%	0.57%	1.10%

Table 7.7: Fault distribution of faults for all merged faults for nodes with overlap period of **1 minute**. *Local%* indicates how many of that node's fault is of this fault type, *Global%* indicates how many of all the node's faults that are this node and fault type, and *Global% of this fault* indicates how many of this fault type occurred at this node. Faults that were combinations of faults are included in all relevant lines. I.e. a ground fault / voltage sag fault is counted both as a voltage sad and a ground fault for that node. 2/2



(a) A wave not related to a fault



(b) A wave 1 minute before a power interruption

Figure 7.4: Sinus waves where there is a sudden change in measured voltage. The red wave is the sub-sample containing the change. The blue wave is the prior sub-sample, and the green is the following.

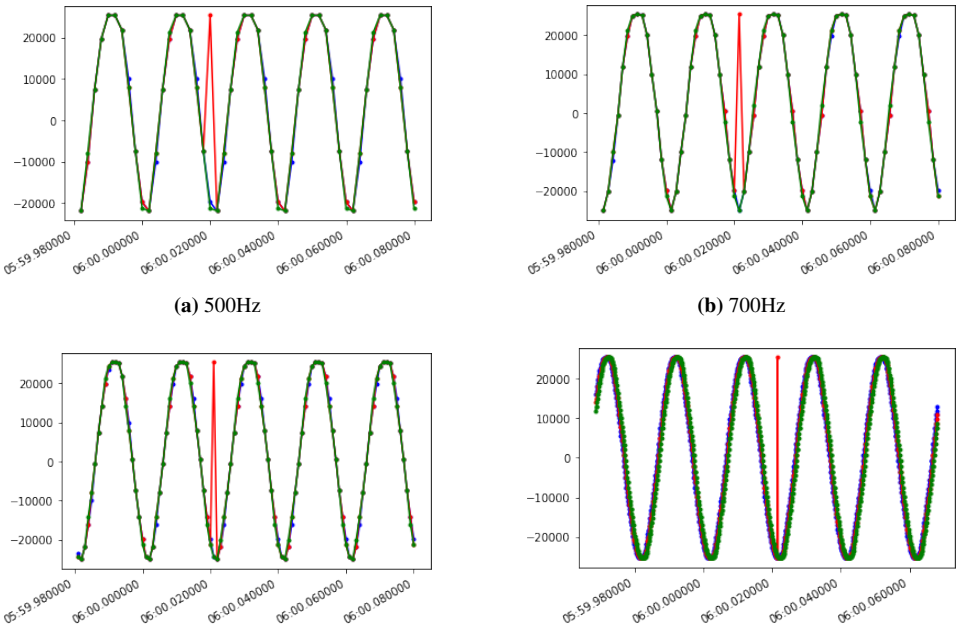
its ability to classify voltage sags on non-300kV nets .

7.2 Inspection of the Waves

We examine different waves to see if we can visually see any changes in the waveform or the frequency of the sinus wave. We do this both per time-step prior to singular faults, and when comparing waves leading up to different fault types, but we cannot see any noticeable differences. As there are no obvious differences between the waves for the different fault types, we suggest that machine learning or other statistical methods should be used to try to differentiate the waves leading up to the different fault types.

7.2.1 Sample Errors

While inspecting the waves we found many sample errors, such as the ones shown in Figure 7.4. Erroneous samples appear to be equal to one of the extremum of the prior wave(s), and were equally frequent in waves without upcoming faults, as waves before power interruptions. The sample errors occurred once in about 75% of the files containing one minute of wave data, rarely more. This leads us to believe that these errors are not related to occurring faults, but rather an error related to the PQA samplers. This might be necessary to take into consideration however, as it might cause peaks – or other errors – in values used in machine learning and statistical methods learning the data, which might become falsely correlated to different fault types.



(a) 500Hz

(b) 700Hz

(c) 1000Hz

(d) 8500Hz

Figure 7.5: Sinus waves where there is a sudden change in measured voltage at different frequencies. The red wave is the sub-sample containing the change. The blue wave is the prior sub-sample, and the green is the following.

7.2.2 Sample Errors and Sampling Frequency Correlation

We examined whether these sampling errors were dependant on sampling frequency, but found no such correlation. An example is shown in Figure 7.5, where there is only one sampling error, occurring at the same time, across different sampling frequencies.

7.3 Clustering

We look at how t-SNE plots look for different data-sets present in Section 6.1, using the aggregation methods presented Section 6.3.3. We inspect t-SNE as this was the dimensionality reduction method that gave the best results in [Jahr and Meen, 2019].

First we inspect how wave form, RMS values and Fourier coefficients compare using t-SNE plots when looking at the combined aggregated values for the data-sets sampled each second for one minute one minute before the faults. The clustering using the wave form shows clear clusters as shown Figure 7.6, but apart from the majority of power interruptions being in one group, the clusters seem to be fairly balanced in what fault types they consist of. The power interruption cluster also appears to contain many non-faults, which might suggest this cluster displays a characteristic of Node2, rather than the fault itself. Using the RMS values does not appear to create any good clusters as shown in Figure 7.7. The clustering using the Fourier coefficients shown in Figure 7.8 creates some lines which appears to be separable from the main cluster, but both the lines and the main cluster appear to be fairly balanced.

When we inspect the data-sets with data sampled one second each minute an hour leading up to the faults, we get the figures shown in Figures 7.9, 7.10, and 7.11. Wave form and RMS values have very similar results to the data-sets sampled each second for one minute one minute before the faults, while Fourier coefficients appear to create a cluster of majority ground faults, as well as one of majority non-faults, aside from its balanced main cluster.

Because of these results, we assume classifiers using the wave form will be the best choice for separating power interruptions from the rest of the data, and RMS values to perform generally worst, as the data does not appear to be easy to separate, which might indicate that classifiers will struggle to differentiate the various faults.

We investigate whether singular values will give better clusters than combined values. The results are shown in Figures 7.12, 7.13, and 7.14. The wave form appears to be rather similar to the plot using combined values, while RMS values and Fourier coefficients clearly fails to form good clusters, only finding an apparent trait of some observations which separates the observations in the middle cluster from the observations of the outer ring. This might suggest that singular values contain too many features, such that the t-SNE method is unable to find any distinct difference or similarity between most of the

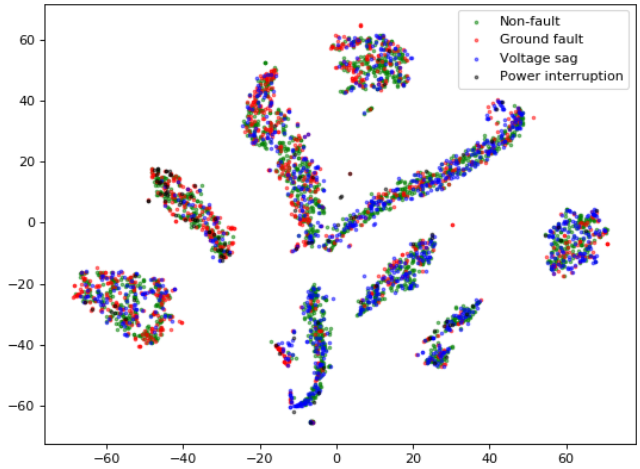


Figure 7.6: t-SNE plot with perplexity 45, using combined aggregated values on the 25kHz wave form data-set presented in Table 6.3.

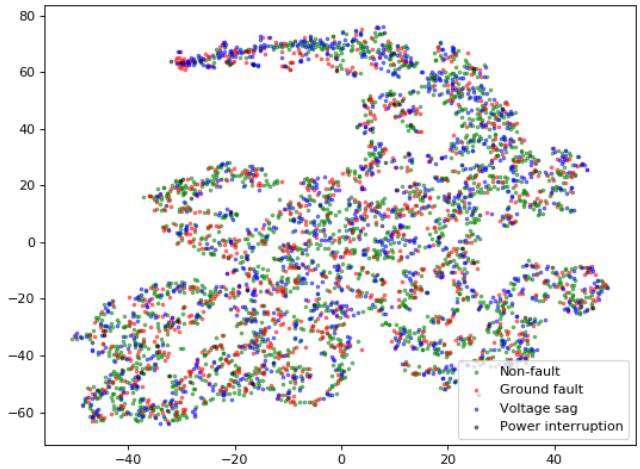


Figure 7.7: t-SNE plot with perplexity 45, using combined aggregated values on the 25kHz RMS value data-set presented in Table 6.5.

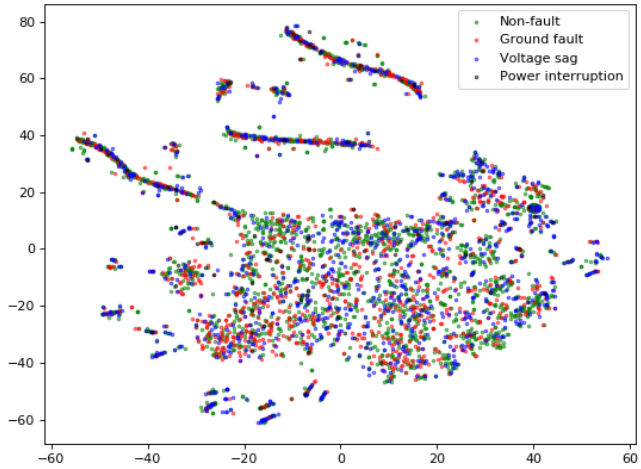


Figure 7.8: t-SNE plot with perplexity 45, using combined aggregated values on the 25kHz Fourier coefficient data-set presented in Table 6.6.

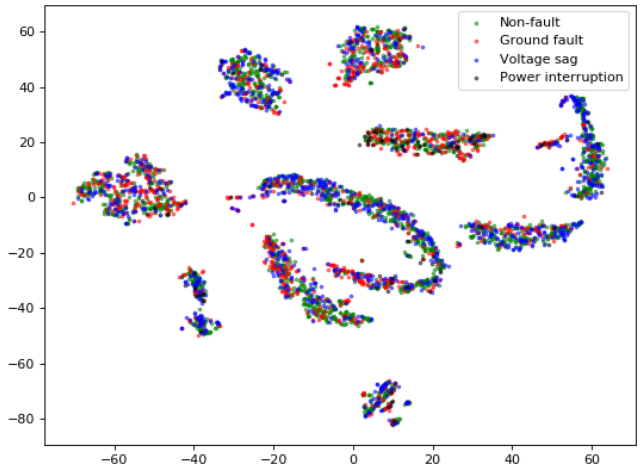


Figure 7.9: t-SNE plot with perplexity 45, using combined aggregated values on the 25kHz wave form data-set presented in Table 6.9.

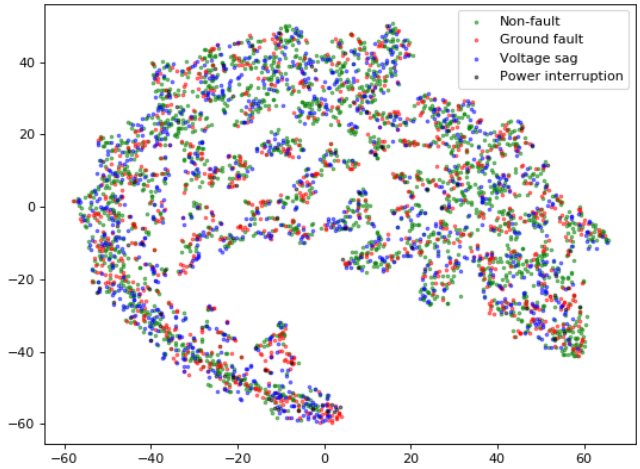


Figure 7.10: t-SNE plot with perplexity 45, using combined aggregated values on the 25kHz RMS value data-set presented in Table 6.11.

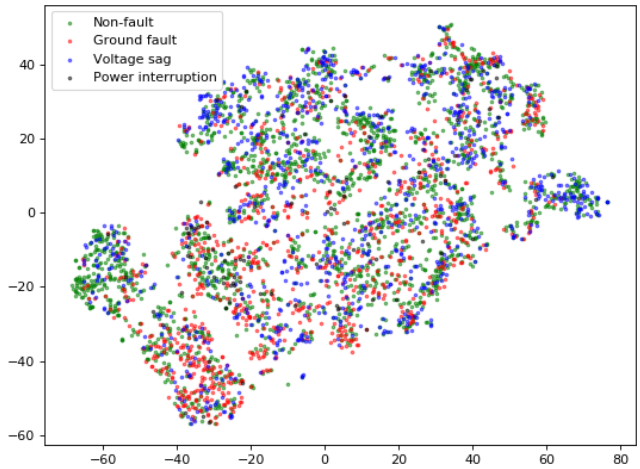


Figure 7.11: t-SNE plot with perplexity 45, using combined aggregated values on the 25kHz Fourier coefficient data-set presented in Table 6.12.

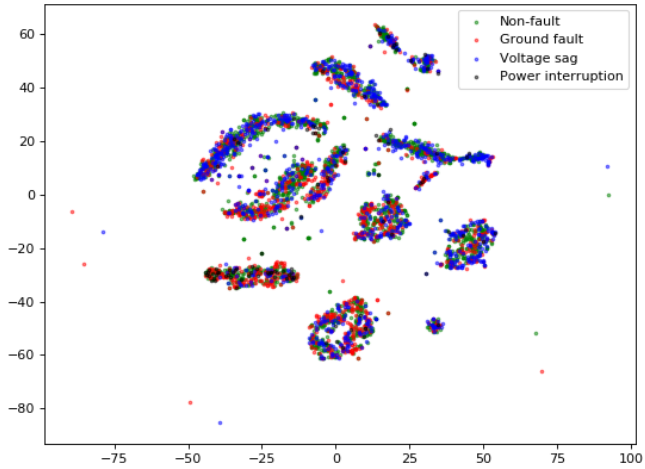


Figure 7.12: t-SNE plot with perplexity 45, using singular aggregated values on the 25kHz wave form data-set presented in Table 6.9.

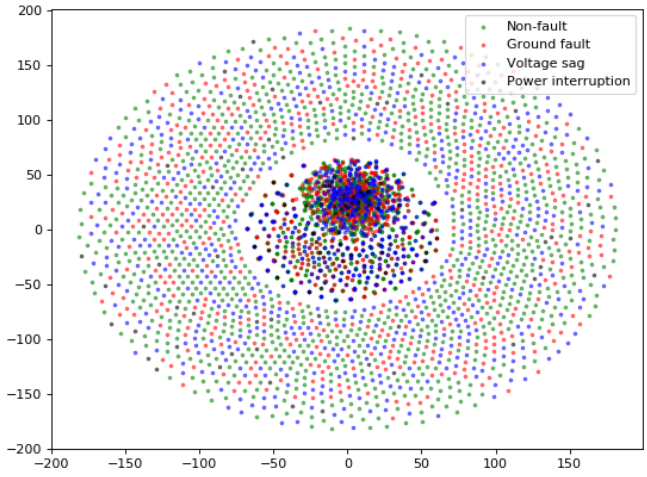


Figure 7.13: t-SNE plot with perplexity 45, using singular aggregated values on the 25kHz RMS value data-set presented in Table 6.11.

observations.

We look at how reducing from 25kHz to 1kHz affects the clustering of the wave form, but Figures 7.15 and 7.16 appear to be just as separable as their 25kHz counterparts.

Lastly we inspect how changing the time before the fault occurs affects the t-SNE plots. The results are shown in Figures 7.17, 7.18, and 7.19. When looking up until zero minutes before the fault, almost all ground faults form one big cluster, suggesting they behave mostly similar and distinctly from other faults. There are some voltage sag observations in the ground fault cluster, but this might be a result of the DDG labeling scheme. Voltage sags mainly populate clusters shared with non-faults, suggesting that they do not contain any signature of the fault at all. Power interruptions surprisingly appear mostly in the ground fault cluster, and the leftmost cluster, mainly consisting of non-faults, which might again suggest that this cluster shows a tendency of Node2 rather than of the fault itself. When we look at one minute and 50 minutes before the fault, the ground fault cluster disappears, and spreads evenly between the other clusters. Also here we see one of the clusters containing most of the power interruptions, in addition to mostly non- and ground faults.

7.4 Line Plots

We inspect how aggregated values change as a value of time, when looking prior to faults occurring, for some of the data-sets presented in Section 6.1. Our goal is to see if we can see the characteristics t-SNE found, and that the classifiers might use to differentiate faults. Some plots being more differentiable than others will also suggest if it is better to focus on some attributes more than others. An example is shown in Figure 7.20. As most of the values are very close we decided to rather look at the 5th, 50th and 95th percentiles, turning the aforementioned figure into Figure 7.21. The other aggregated values are shown in Figure 7.22. Once again power interruptions stand out as an outlier, but apart from that there appear to be no noticeable changes in the aggregated values prior to the fault occurring, where the min and max values change significantly. STD and SNR appear to be somewhat different for the different fault types, and might be differentiable for classifiers.

Looking at the min aggregation or the max aggregation for V2 and V3 – shown in Figures 7.23, 7.24, and 7.25 respectively – reveals no additional information. We also compare the wave form, RMS value, and Fourier coefficient plots, presented in Figures 7.26, 7.27, and 7.28 respectively. Aside from power interruptions being an outlier, and slightly different ceilings for the max values for the different types, do we not see anything notable, akin to what we found for Figure 7.22.

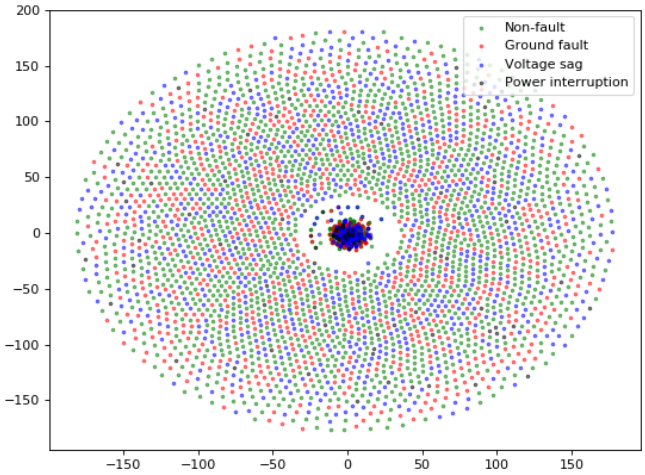


Figure 7.14: t-SNE plot with perplexity 45, using singular aggregated values on the 25kHz Fourier coefficient data-set presented in Table 6.12.

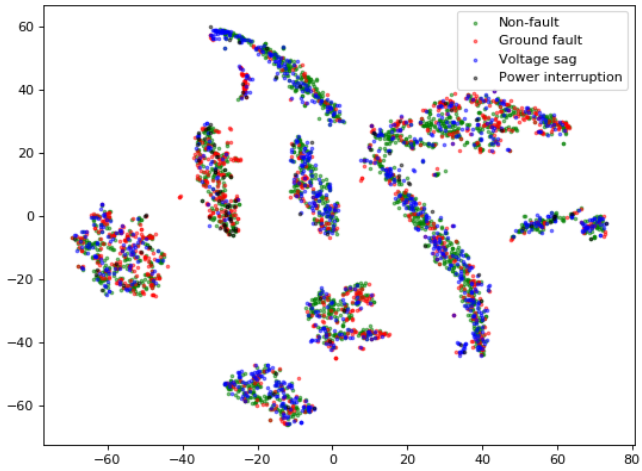


Figure 7.15: t-SNE plot with perplexity 45, using combined aggregated values on the 1kHz wave form data-set presented in Table 6.1.

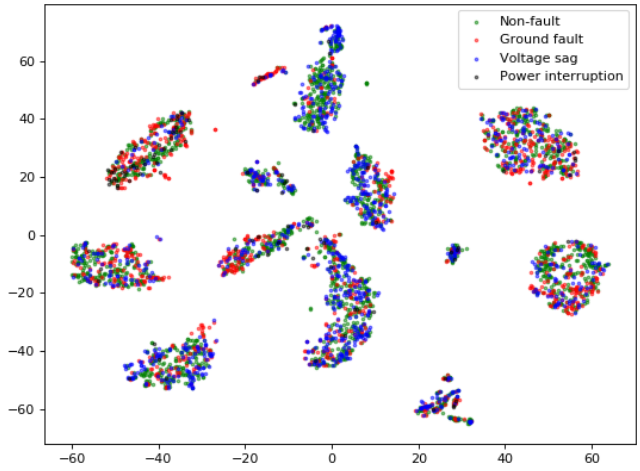


Figure 7.16: t-SNE plot with perplexity 45, using combined aggregated values on the 1kHz wave form data-set presented in Table 6.7.

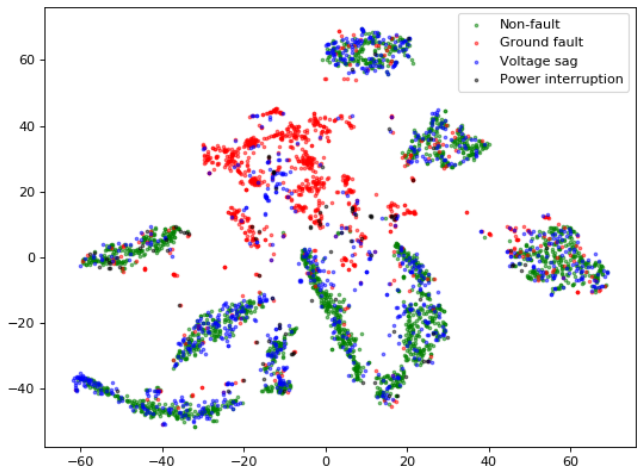


Figure 7.17: t-SNE plot with perplexity 45, using combined aggregated values on the 0 minutes before fault 1kHz wave form data-set presented in Table 6.13.

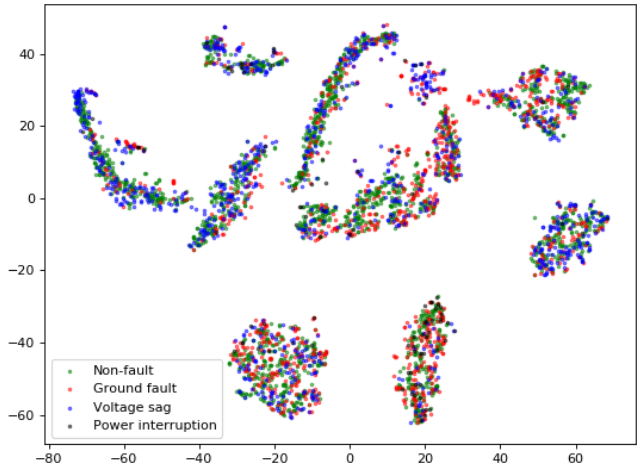


Figure 7.18: t-SNE plot with perplexity 45, using combined aggregated values on the 1 minute before fault 1kHz wave form data-set presented in Table 6.14.

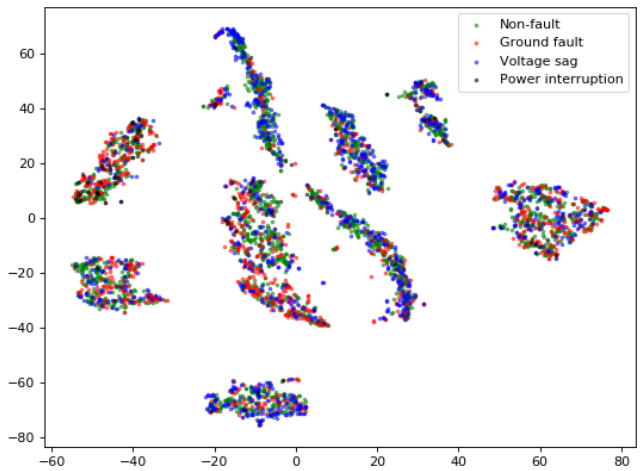


Figure 7.19: t-SNE plot with perplexity 45, using combined aggregated values on the 50 minutes before fault 1kHz wave form data-set presented in Table 6.19.

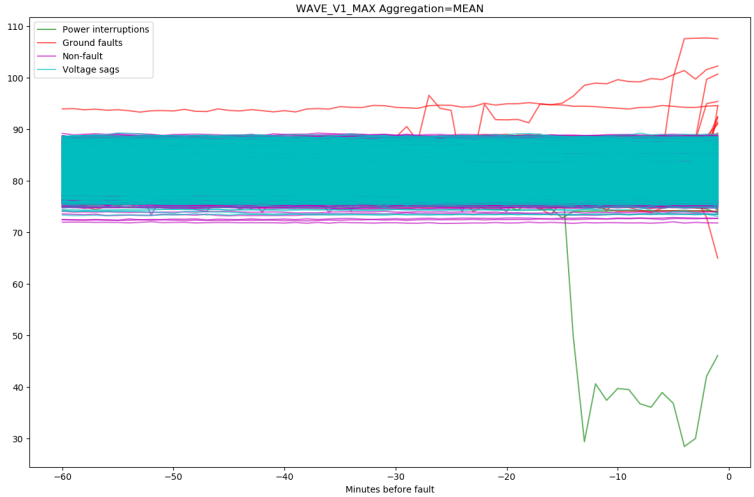


Figure 7.20: The aggregated mean given from the V1 max aggregation method on the 0 minutes before fault 1kHz wave form data-set presented in Table 6.13.

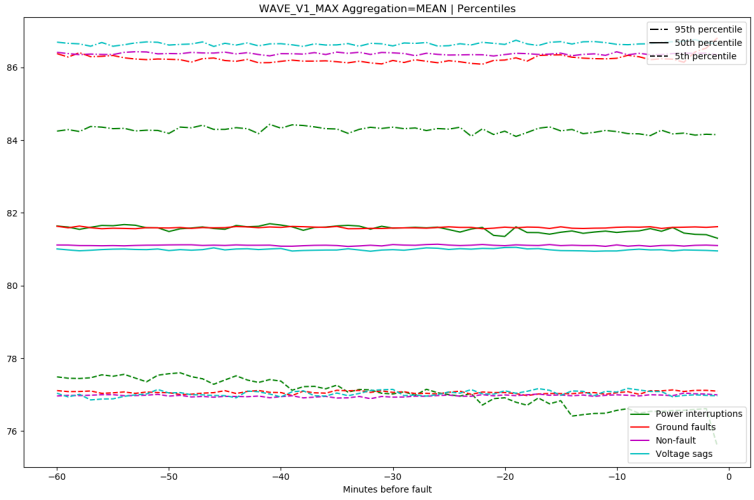
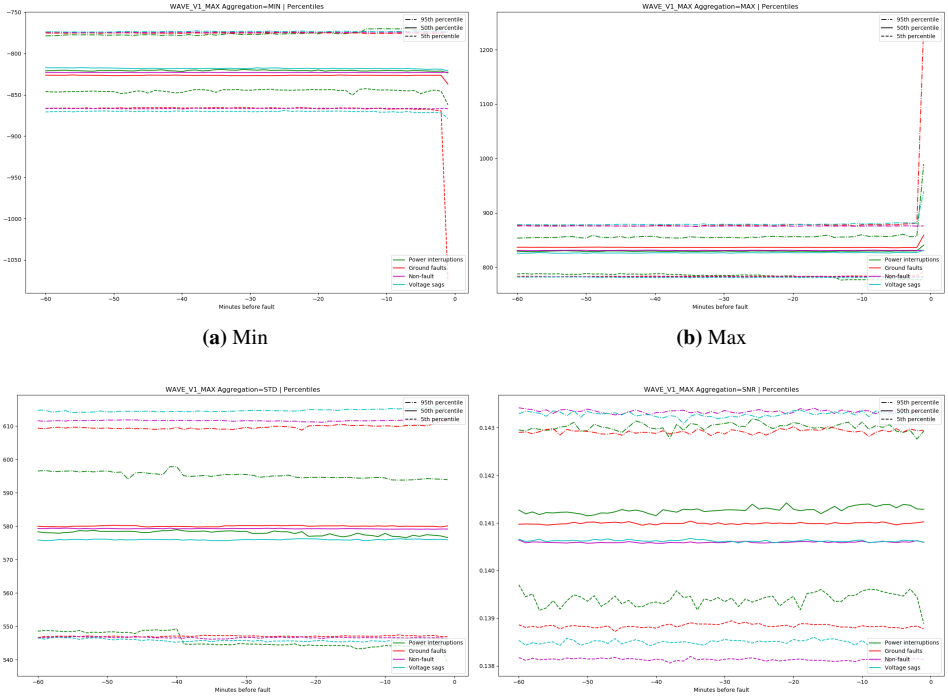


Figure 7.21: The 5th, 50th and 95th percentile of the aggregated mean given from the V1 max aggregation method on the 0 minutes before fault 1kHz wave form data-set presented in Table 6.13.



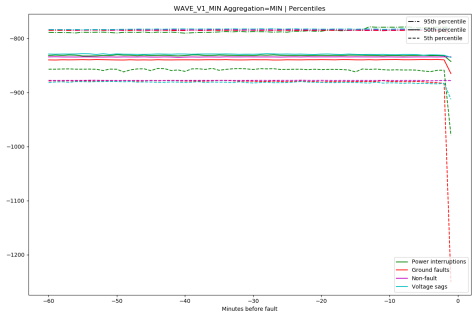
(a) Min

(b) Max

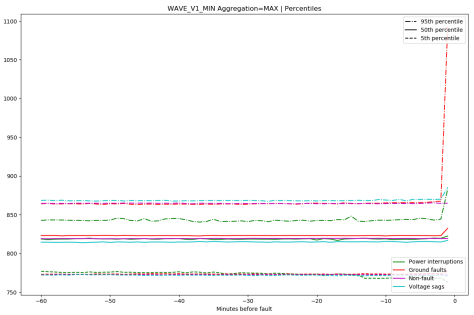
(c) STD

(d) SNR

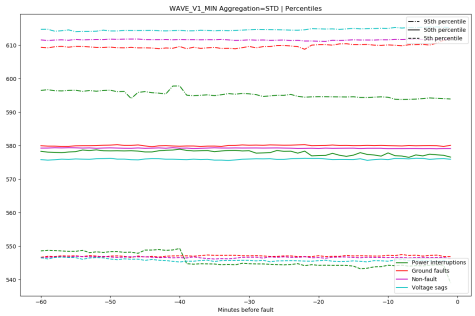
Figure 7.22: The 5th, 50th and 95th percentile of various aggregated values given from the V1 max aggregation method on the 0 minutes before fault 1kHz wave form data-set presented in Table 6.13.



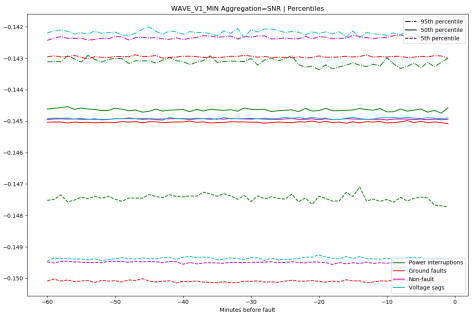
(a) Min



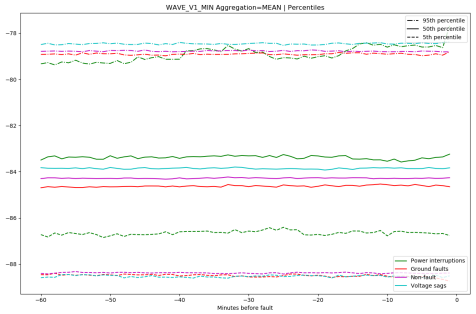
(b) Max



(c) STD

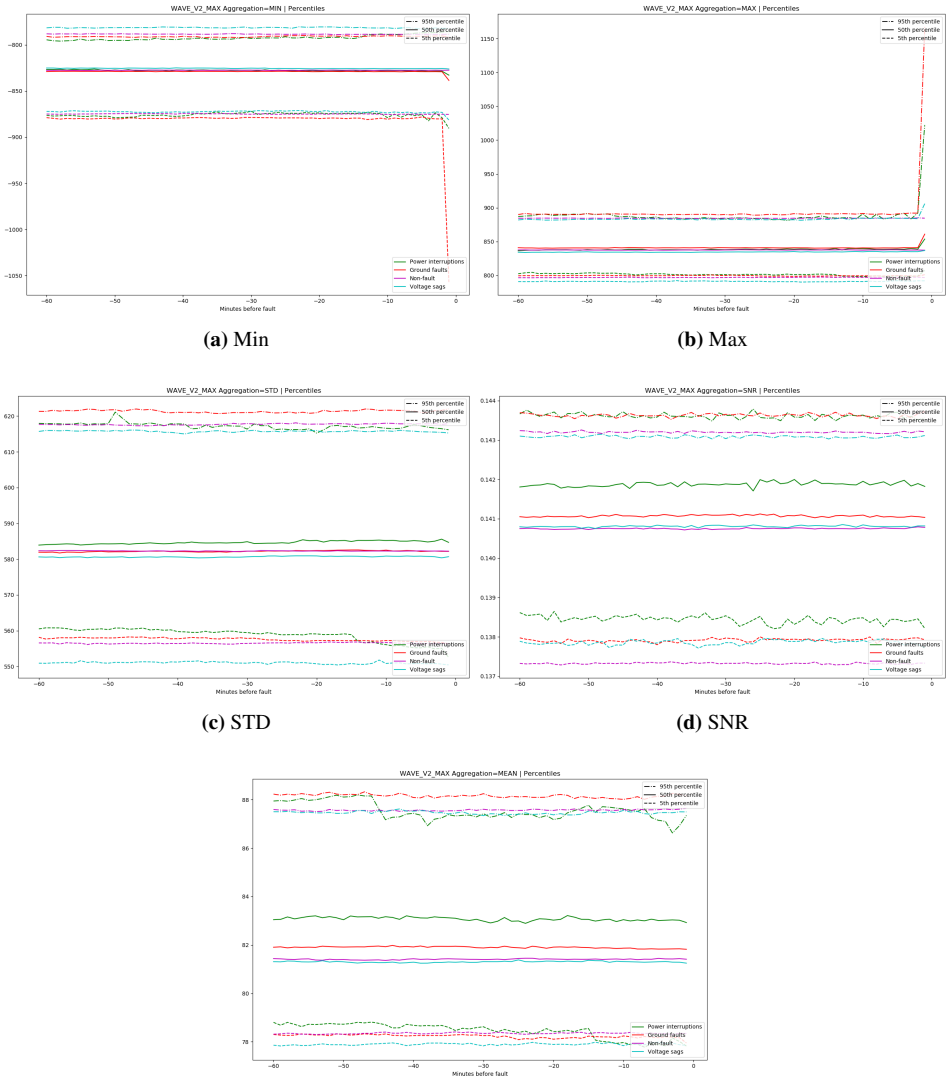


(d) SNR



(e) Mean

Figure 7.23: The 5th, 50th and 95th percentile of various aggregated values given from the V1 min aggregation method on the 0 minutes before fault 1kHz wave form data-set presented in Table 6.13.



(a) Min

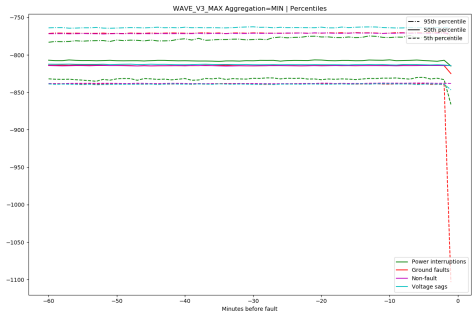
(b) Max

(c) STD

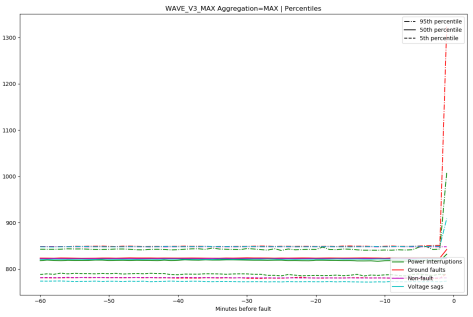
(d) SNR

(e) Mean

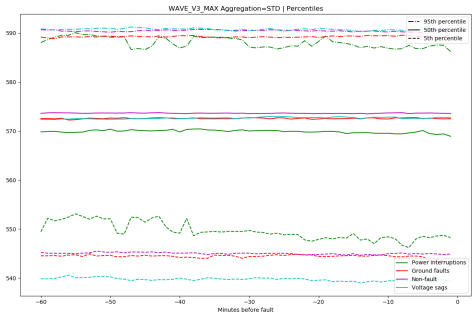
Figure 7.24: The 5th, 50th and 95th percentile of various aggregated values given from the V2 max aggregation method on the 0 minutes before fault 1kHz wave form data-set presented in Table 6.13.



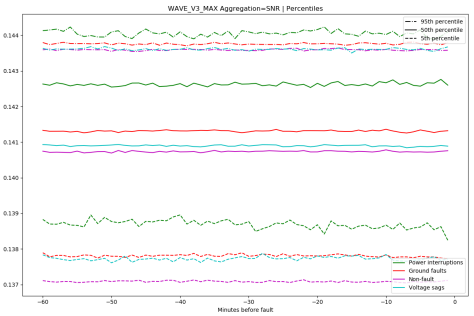
(a) Min



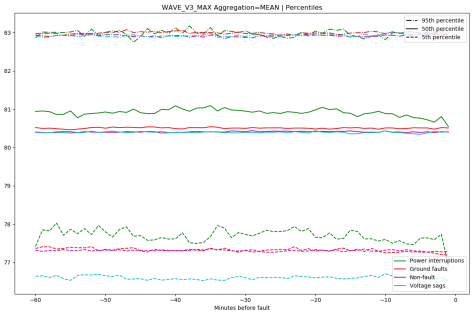
(b) Max



(c) STD

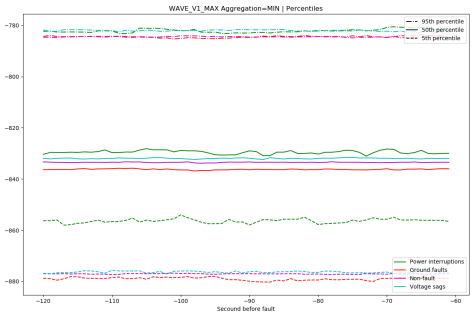


(d) SNR

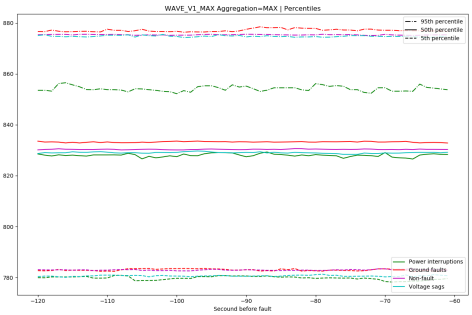


(e) Mean

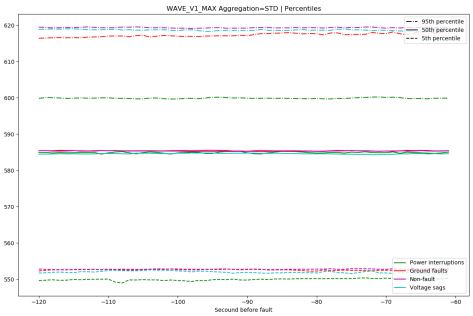
Figure 7.25: The 5th, 50th and 95th percentile of various aggregated values given from the V3 max aggregation method on the 0 minutes before fault 1kHz wave form data-set presented in Table 6.13.



(a) Min



(b) Max

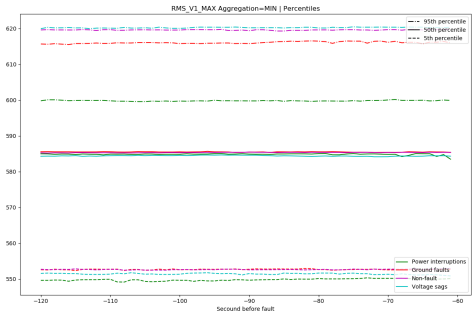


(c) STD

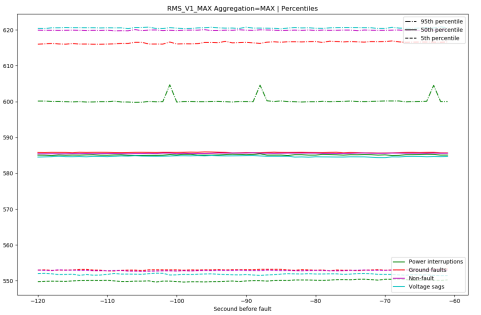
(d) SNR

(e) Mean

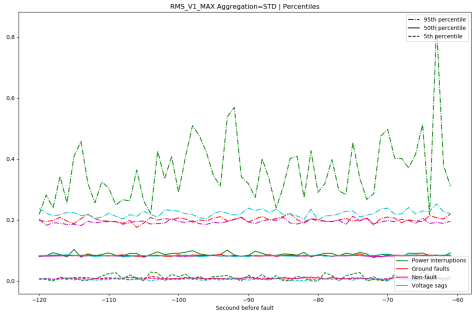
Figure 7.26: The 5th, 50th and 95th percentile of various aggregated values given from the V1 max aggregation method on the 25kHz wave form data-set presented in Table 6.3.



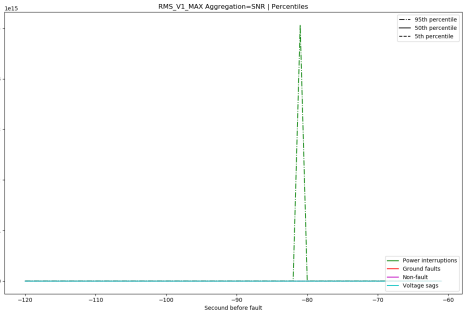
(a) Min



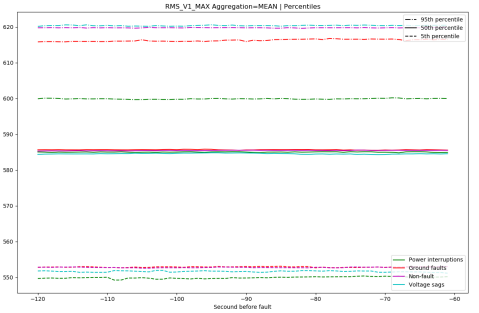
(b) Max



(c) STD

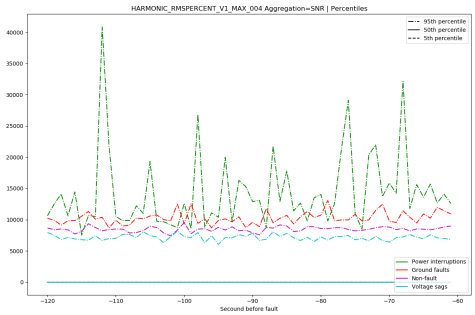


(d) SNR

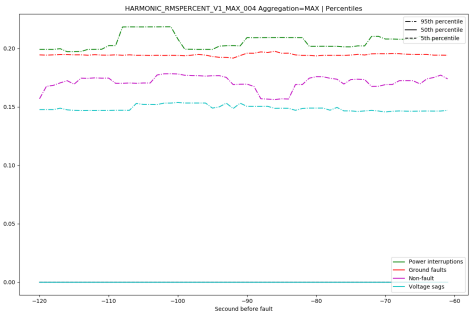


(e) Mean

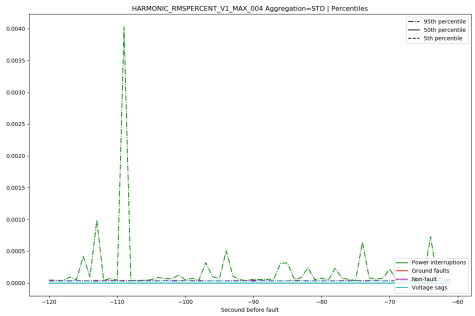
Figure 7.27: The 5th, 50th and 95th percentile of various aggregated values given from the V1 max aggregation method on the 25kHz RMS value data-set presented in Table 6.5.



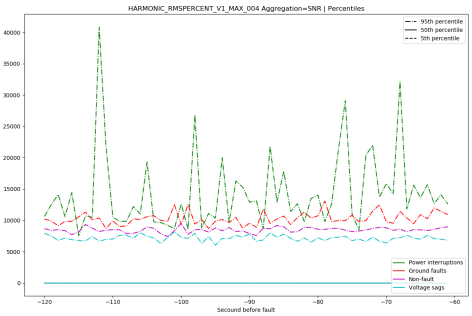
(a) Min



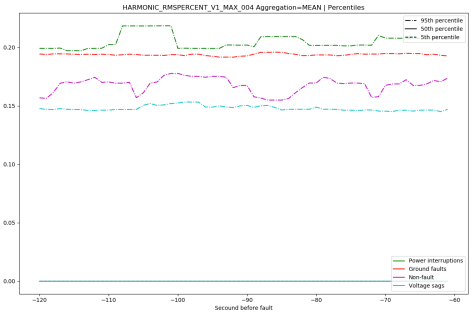
(b) Max



(c) STD



(d) SNR



(e) Mean

Figure 7.28: The 5th, 50th and 95th percentile of various aggregated values given from the V1 max aggregation method on the 25kHz Fourier coefficient data-set presented in Table 6.6.

7.5 Distribution of Nodes

We take note of the frequency of some power interruptions clustering together with non-faults in the t-SNE plots, and the imbalance discussed in Section 7.1.4. One possible explanation for this tendency could be that the t-SNE finds characteristics of the various nodes, not of the faults themselves. We therefore try to label the t-SNE plots based on the origin node, rather than the fault types. As t-SNE is unsupervised, this would reveal if the differences between the various nodes are greater than the differences between fault types within the nodes, and if using unsupervised models on this data serve any purpose other than finding underlying structure between the nodes.

We recreate Figures 7.6, 7.7, and 7.8 as Figures 7.29, 7.30, and 7.31, with nodes as labels. It is clear that what t-SNE finds to be similar between observations are mainly the nodes, not the fault type, especially for the wave form. The tendency is also present in the RMS value and Fourier coefficient plots, albeit to a lesser degree. The reason the effect is most prominent in the wave form data-sets is likely due to RMS values and Fourier coefficients being approximations of the waves rather than the wave itself, removing some characteristics of the wave in their approximations. For wave form the vast majority of observations are grouped primarily with observations from the same node, seemingly independent of the voltage used by the node. We recreate the zero minutes before fault plot shown in Figure 7.17, which is the plot with the highest likelihood of successfully separating based on fault types. The recreation, shown in Figure 7.32, shows that while there are more outliers, mainly in the ground fault cluster of the former, the vast majority of observations are still more differentiable when looking at their node rather than their fault type, including the power interruptions being a part of the Node2 cluster as previously suspected.

7.5.1 Clustering for Each Node

We attempt to recreate the clusters of the different faults for each node using t-SNE. We again recreate the zero minutes before fault plot shown in Figure 7.17. This recreation, displayed as 12 separate t-SNE plots, are shown in Figures 7.33 and 7.34. Here we again see ground faults clearly separating from non-faults, and when we look at Node2, power interruptions appear to group more with ground fault than other types. This indicates that there are clear differences between the different fault types, even though they are less prominent than the differences between different nodes. We recreate Figures 7.6, 7.7, and 7.8 as Figures 7.35, 7.36, and 7.37, but we cannot see any clustering of fault types, for any of the aggregation methods or nodes, indicating that the differences between the underlying structures of the fault types are slim when we look at periods not including the start of the fault.

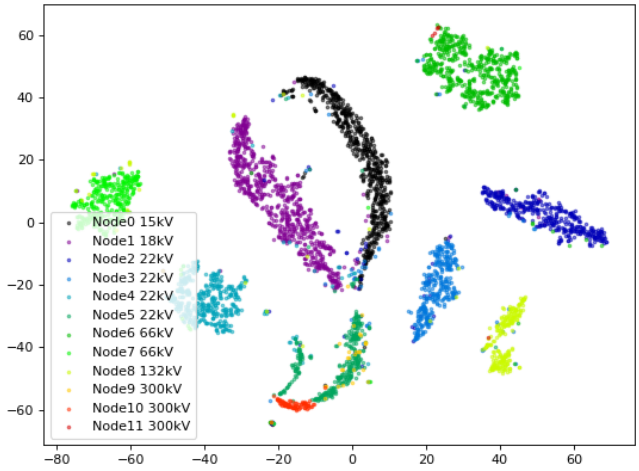


Figure 7.29: Recreation of Figure 7.6 with nodes as labels.
t-SNE plot with perplexity 45, using combined aggregated values on the 25kHz wave form data-set presented in Table 6.3.

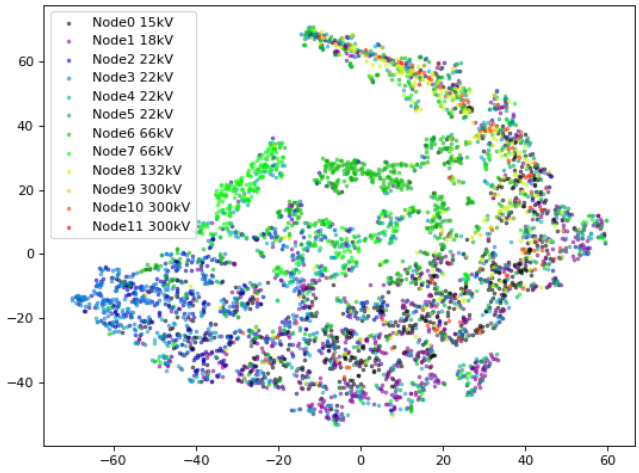


Figure 7.30: Recreation of Figure 7.7 with nodes as labels.
t-SNE plot with perplexity 45, using combined aggregated values on the 25kHz RMS value data-set presented in Table 6.5.

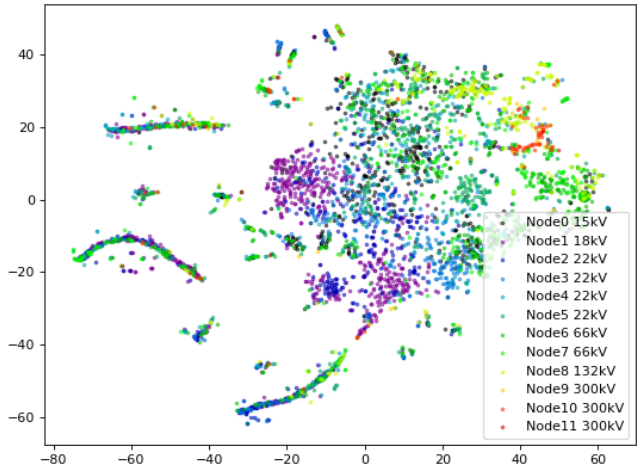


Figure 7.31: Recreation of Figure 7.8 with nodes as labels.
t-SNE plot with perplexity 45, using combined aggregated values on the 25kHz Fourier coefficient data-set presented in Table 6.6.

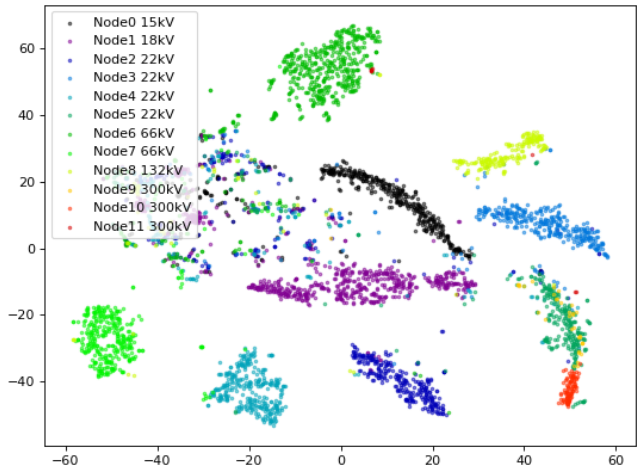
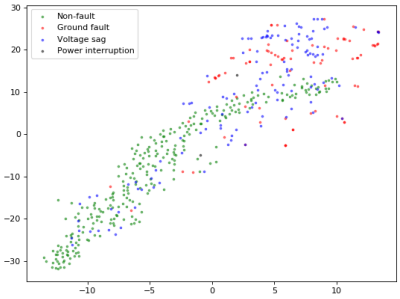
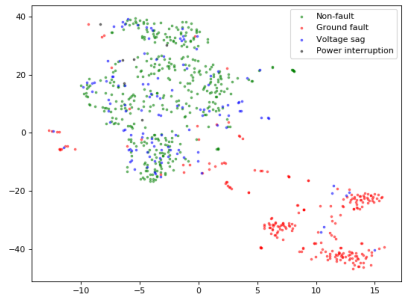


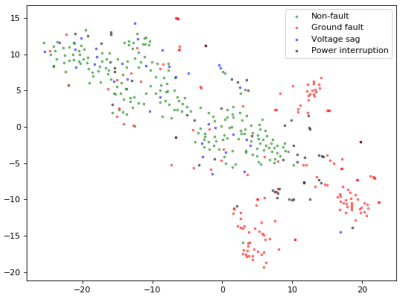
Figure 7.32: Recreation of Figure 7.17 with nodes as labels.
t-SNE plot with perplexity 45, using combined aggregated values on the 0 minutes before fault 1kHz wave form data-set presented in Table 6.13.



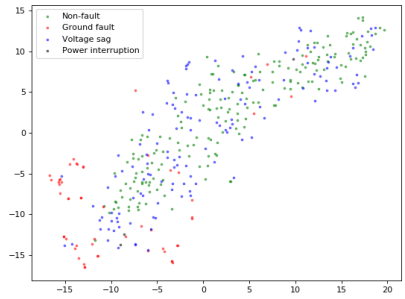
(a) Node0



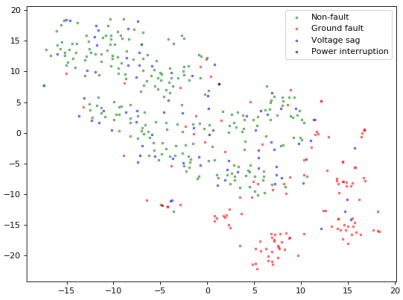
(b) Node1



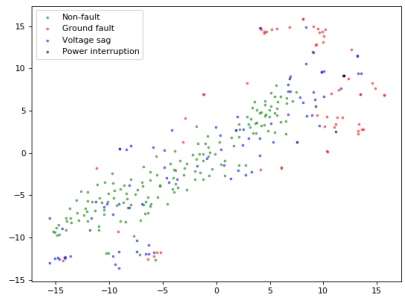
(c) Node2



(d) Node3

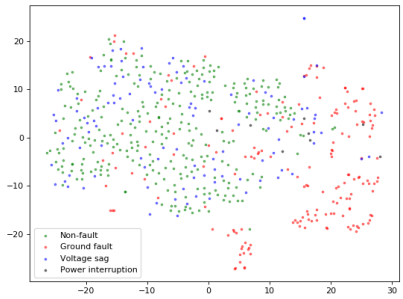


(e) Node4

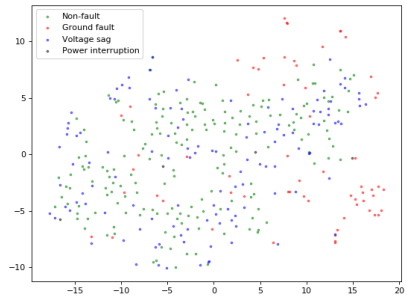


(f) Node5

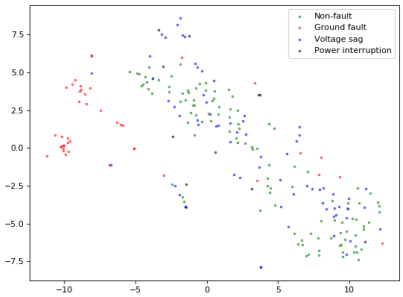
Figure 7.33: Recreation of Figure 7.17 with t-SNE for each individual node.
t-SNE plot with perplexity 45, using combined aggregated values on the 0 minutes before fault 1kHz wave form data-set presented in Table 6.13. 1/2



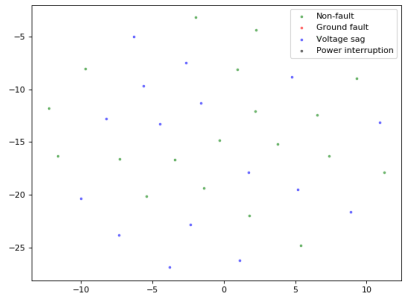
(a) Node6



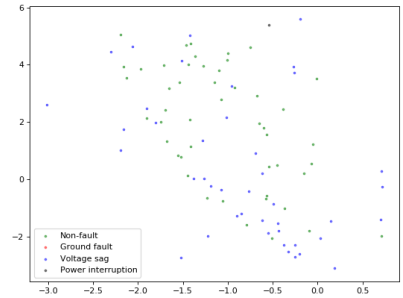
(b) Node7



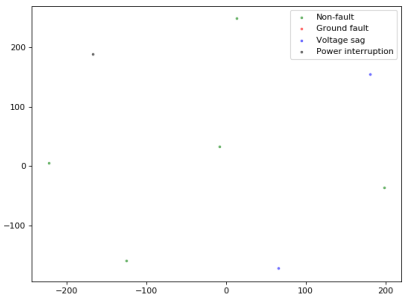
(c) Node8



(d) Node9

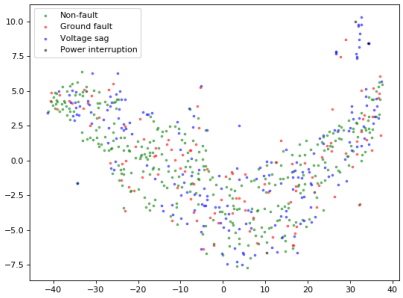


(e) Node10

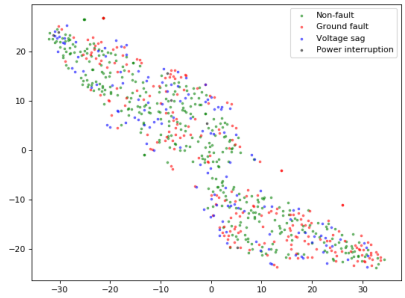


(f) Node11

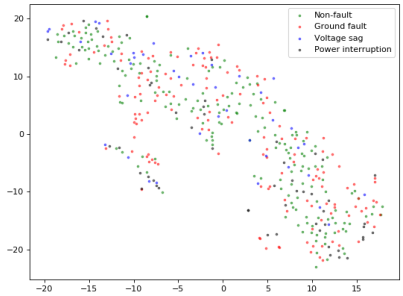
Figure 7.34: Recreation of Figure 7.17 with t-SNE for each individual node.
t-SNE plot with perplexity 45, using combined aggregated values on the 0 minutes before fault 1kHz wave form data-set presented in Table 6.13. 2/2



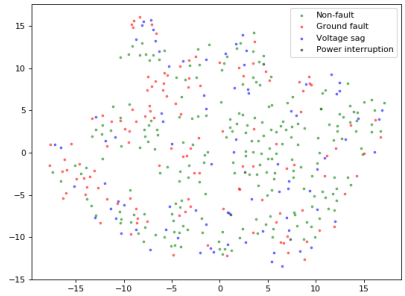
(a) Node0



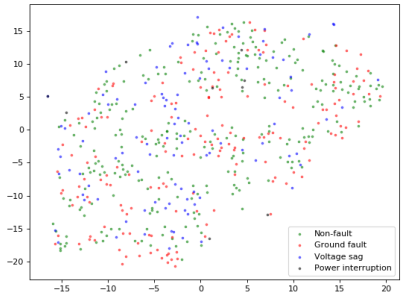
(b) Node1



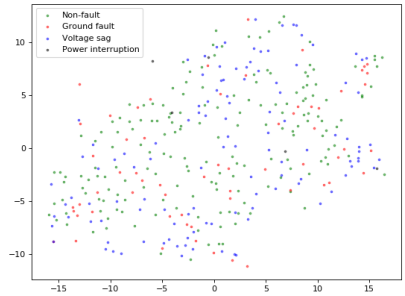
(c) Node2



(d) Node4

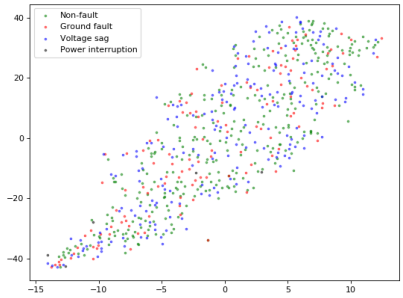


(e) Node6

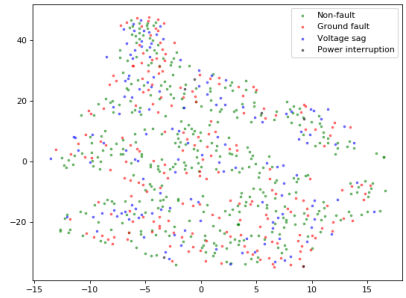


(f) Node7

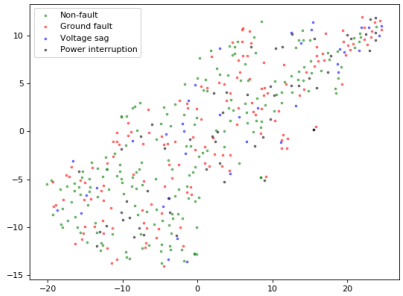
Figure 7.35: Recreation of Figure 7.6 with t-SNE for a selection of individual nodes.
t-SNE plot with perplexity 45, using combined aggregated values on the 25kHz wave form data-set presented in Table 6.3.



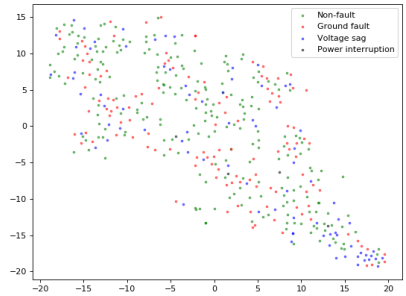
(a) Node0



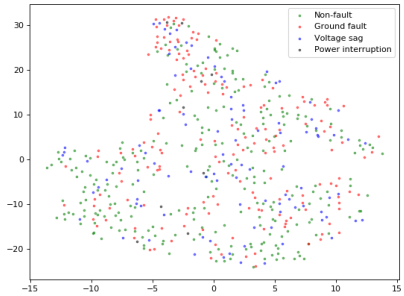
(b) Node1



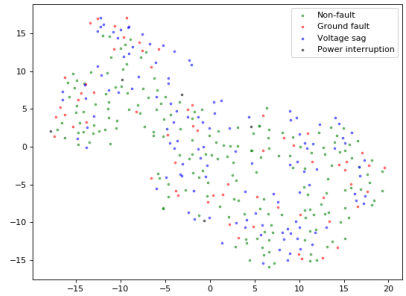
(c) Node2



(d) Node4

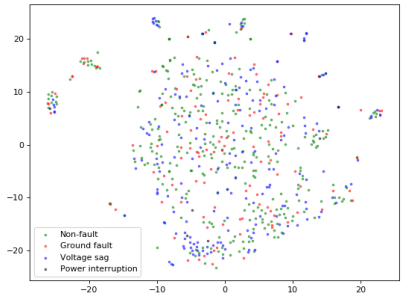


(e) Node6

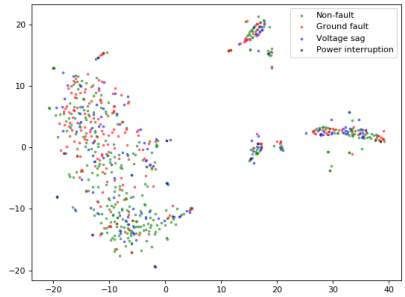


(f) Node7

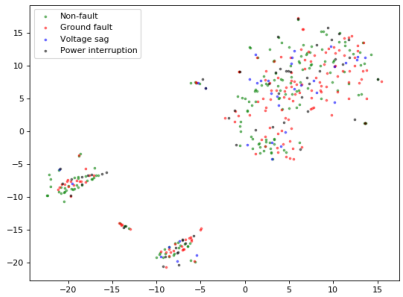
Figure 7.36: Recreation of Figure 7.7 with t-SNE for a selection of individual nodes.
t-SNE plot with perplexity 45, using combined aggregated values on the 25kHz RMS value data-set presented in Table 6.5.



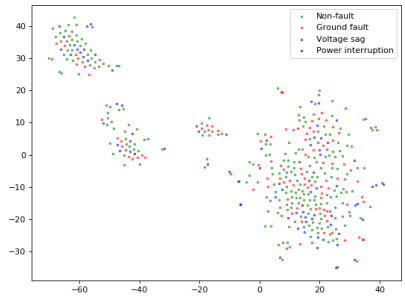
(a) Node0



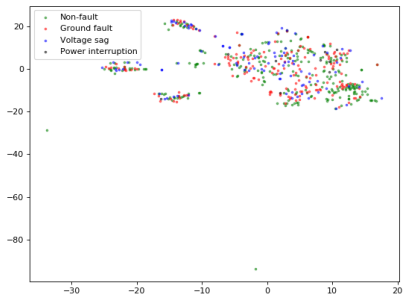
(b) Node1



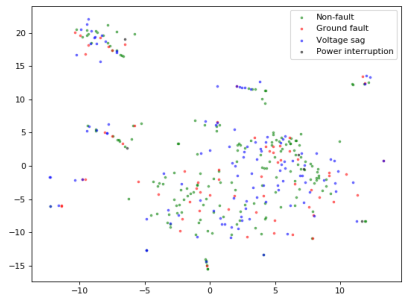
(c) Node2



(d) Node4



(e) Node6



(f) Node7

Figure 7.37: Recreation of Figure 7.8 with t-SNE for a selection of individual nodes.
t-SNE plot with perplexity 45, using combined aggregated values on the 25kHz Fourier coefficient data-set presented in Table 6.6.

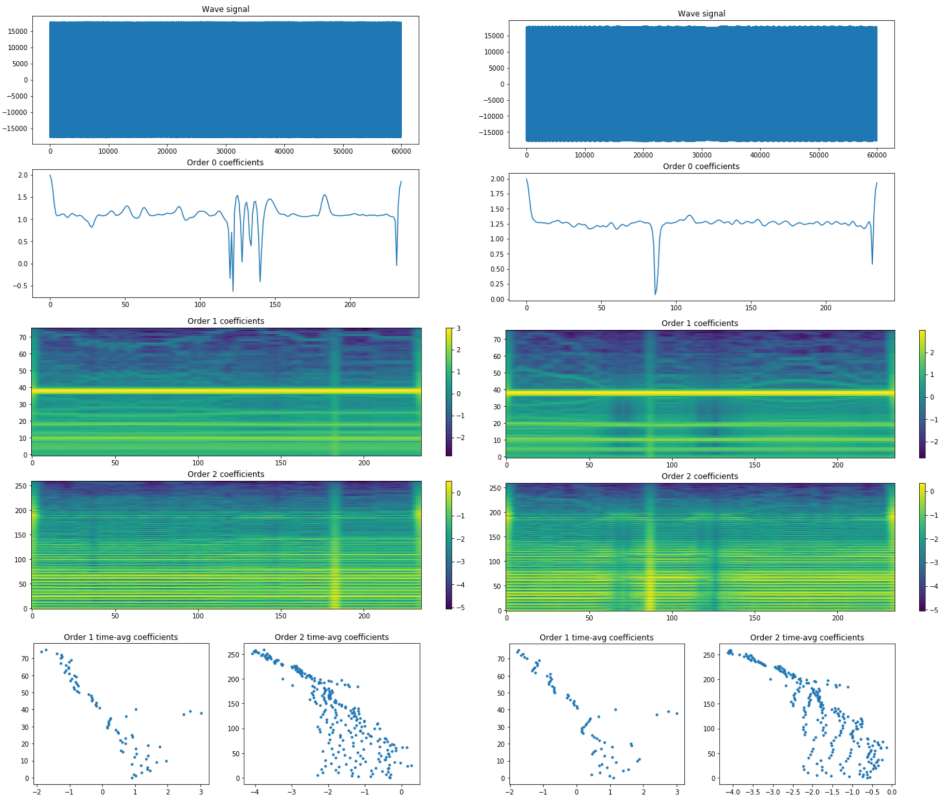
7.6 Wavelets

7.6.1 Wavelet Scattering

We inspect how the wavelet scattering coefficients of faults and non-faults look like and if there are any noticeable differences that are visible to the human eye. An illustration of the wavelet scattering coefficients of a ground fault and a non-fault from the same node can be seen in Figure 7.38. The bright yellow line visible at the order 1 coefficients is 50Hz. There are some visible differences in all levels. These differences might not necessarily be traits of the fault, they could be variations common between all signals or just noise, but it looks like there is some potential here. The time-averaged coefficients for both the final levels do not seem to differ significantly, which might suggest that the differences seen are common variations or noise.

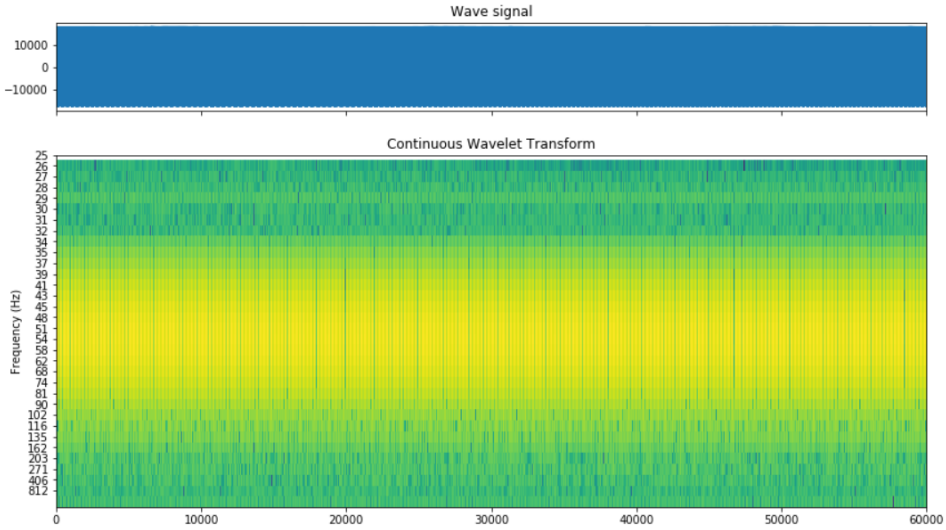
7.6.2 Wavelet Transform Spectograms

We inspect how the wavelet transform spectograms of faults and non-faults look like and if there are any noticeable differences that are visible to the human eye. An illustration of the continuous wavelet transform of a ground fault and a non-fault from the same node can be seen in Figure 7.39. The data used was the same as in Figure 7.38. We can see that most of the power is centered around 50Hz, but as in Figure 7.38 with wavelet scattering there are no obvious differences in the structure between the ground fault and the non-fault. Considering that we only are looking at a one minute interval one minute before the fault occurs, this might suggest that there are no traits of the fault appearing in this interval that differentiates it from the non-fault, and if they appear they are very weak or not extractable by the wavelet transform.

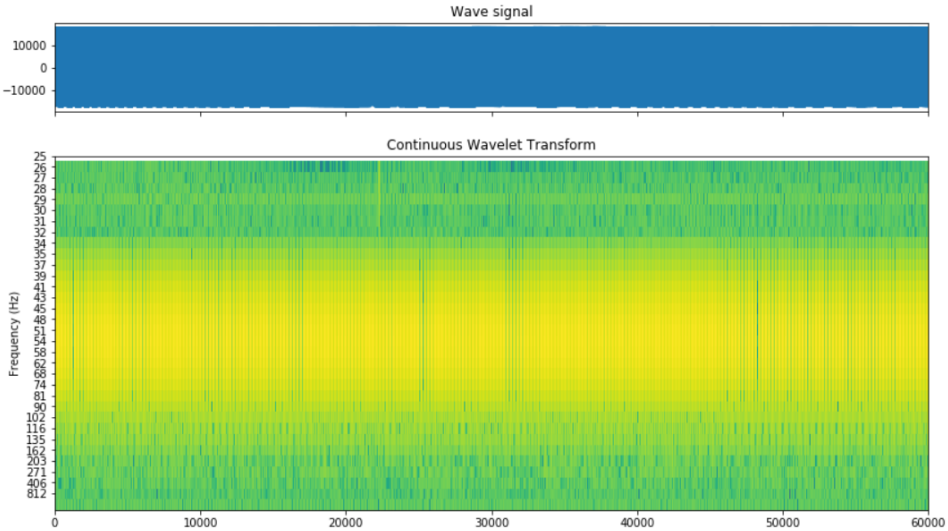


(a) Ground fault **(b)** Non-fault

Figure 7.38: Wavelet scattering coefficients for the first three levels of a ground fault (a) and a non-fault (b) sampled from the same node. The data used is the 1kHz wave form data-set presented in Table 6.1, only the V1 max aggregation was used. For the wavelet scattering the parameters used were J=8 and Q=12. For the order 1 and order 2 coefficients the time-averaged coefficients are also plotted, i.e. they are averaged over the x-axis which is the time-axis.



(a) Ground fault



(b) Non-fault

Figure 7.39: Spectograms of the continuous wavelet transform of a ground fault (a) and a non-fault (b) sampled from the same node. The data used is the 1kHz wave form data-set presented in Table 6.1, only the V1 max aggregation was used. For the transform the scales used were from 1 to 32 and the mother wavelet used was the Morlet wavelet.

Chapter 8

Experiments

In this chapter we will introduce the experiments we will carry out on the data presented in Chapter 6.

8.1 Classifiers

The classifiers used in the experiments in this thesis use their default parameters, apart from the parameters presented in Table 8.1. SVM, *k*-NN, and random forest were implemented using sklearn [Pedregosa et al., 2011], CatBoost is a stand alone project at catboost.ai [Prokhorenkova et al., 2017], CNN was implemented using pytorch [Paszke et al., 2019] and FFNN was implemented using keras [Chollet et al., 2020].

8.2 Experiments

8.2.1 Experiment 1

In this experiment we wish to explore whether aggregating the pure wave data give comparable results to aggregating Fourier coefficients of the wave, or aggregating the RMS values of the wave.

Method	**Parameters**
SVM	Kernel = linear, sigmoid, rbf, poly Degree (poly only) = 2, 3, 4, 5, 6 Probability = True Max iterations = 50,000
RF	Estimators = 50, 100, 150, 200, 250, 300, 350, 400, 450, 500 Max depth = 5, 10, 15, 20, 25, 30, None
CatBoost	Eval metric = AUC Iterations = 1000 Learning rate = 0.1 Early stopping = 150
k-NN	k = 3
FFNN	Layer 1 size = 64 (Aggregated values) 1024 (Wavelets) Layer 2 size = 32 (Aggregated values) 512 (Wavelets) Layer 3 size = 16 (Aggregated values) 256 (Wavelets) Layer 4 as output layer Activation function = ReLU Optimizer = Adam Learning rate = 0.001 Loss function = Cross Entropy Loss Batch size = 128 (Aggregated values) 32 (Wavelets) Number of epochs = 300 (Aggregated values) 150 (Wavelets)
CNN	Convolution layer 1 size = 32 Convolution layer 1 kernel size = 5×5 Pooling layer 1 type = maximum Pooling layer 1 kernel size = 2×2 Convolution layer 2 output = 64 Convolution layer 2 kernel size = 3×3 Fully connected layer 1 size = 256 or 4096 Fully connected layer 2 size = 84 or 512 Fully connected layer 3 as output layer Activation layer function = ReLU Optimizer = Adam Learning rate = 0.001 Loss function = Cross Entropy Loss Batch size = 32 Number of epochs = 150

Table 8.1: Classifiers with their parameters.

Data

The 25kHz data-sets shown in Tables 6.3, 6.5, and 6.6, using the combined aggregated values.

Note

There was a bug with DDG which caused it to retrieve 20kHz rather than 25kHz for the Fourier coefficient data-set given in Table 6.6, this was discovered too late to make new data, and as so we have used the 20kHz data in place of the intended 25kHz data. We do not think this error invalidates our results, but it should be noted that Fourier coefficients might have reached slightly different results if it was sampled at the intended frequency.

Method

For each data-set, train a SVM, a random forest, a FFNN, and a CatBoost model using the parameters in Table 8.1, compare the results.

8.2.2 Experiment 2

In this experiment we wish to explore whether aggregating the pure wave data using singular aggregated values gives comparable results to Experiment 1.

Data

The 25kHz data-sets shown in Tables 6.3, 6.5, and 6.6, using both singular and combined aggregated values.

Method

For each data-set, train a SVM, a random forest, a FFNN, and a CatBoost model using the parameters in Table 8.1, compare the results.

8.2.3 Experiment 3

In this experiment we wish to explore whether looking at data-sets with data sampled one second each minute an hour leading up to the faults, gives comparable results to Experiment 1 and 2.

Data

The 25kHz data-sets shown in Tables 6.3, 6.5, and 6.6, and the data-sets in Tables 6.9, 6.11, and 6.12, using both singular and combined aggregated values.

Method

For each data-set, train a SVM, a random forest, a FFNN, and a CatBoost model using the parameters in Table 8.1, compare the results.

8.2.4 Experiment 4

In this experiment we wish to explore whether using wavelet scattering gives comparable results to Experiment 1, 2, and 3.

Data

The 25kHz wave form data-set shown in Table 6.3, using wavelet scattering and both combined and singular aggregated values. For wavelet scattering only the V1 max aggregation was used to limit the scope.

Method

Train a CatBoost model, a k-NN, a FFNN for wavelet scattering using the parameters in Table 8.1, compare the results.

8.2.5 Experiment 5

In this experiment we wish to explore the importance of sampling frequency, and if it is easier to distinguish faults using a signal sampled at a high frequency compared to a signal sampled at a low frequency.

Data

The wave form data-sets shown in Tables 6.1, 6.2, 6.3, and 6.4, using the combined aggregated values.

Method

For each data-set, train a SVM, a random forest, a FFNN, and a CatBoost model using the parameters in Table 8.1, compare the results.

8.2.6 Experiment 6

In this experiment we wish to explore whether aggregating the different frequencies using singular aggregated values gives comparable results to Experiment 5.

Data

The wave form data-sets shown in Tables 6.1, 6.2, 6.3, and 6.4, using both singular and combined aggregated values.

Method

For each data type, train a SVM, a random forest, a FFNN, and a CatBoost model using the parameters in Table 8.1, compare the results.

8.2.7 Experiment 7

In this experiment we wish to explore whether looking at data-sets with data sampled one second each minute an hour leading up to the faults, gives comparable results to Experiment 5 and 6.

Data

The wave form data-sets shown in Tables 6.1, 6.2, 6.3, and 6.4, 6.7, 6.8, 6.9 and 6.10, using both singular and combined aggregated values.

Method

For each data-set, train a SVM, a random forest, a FFNN, and a CatBoost model using the parameters in Table 8.1, compare the results.

8.2.8 Experiment 8

In this experiment we wish to explore whether using wavelet scattering gives comparable results to Experiment 5, 6, and 7.

Data

The wave form data-set shown in Tables 6.1 and 6.3, using wavelet scattering and both combined and singular aggregated values.

Method

Train a CatBoost model, a k-NN, a FFNN for wavelet scattering using the parameters in Table 8.1, compare the results.

8.2.9 Experiment 9

In this experiment we wish to examine how prediction results change when we look at the wave at different times before the fault occurs.

Data

The wave form data-set shown in Tables 6.13, 6.14, 6.15, 6.17, 6.18, and 6.19, using combined aggregated values.

Method

For each data type, train a SVM, a random forest, a FFNN, and a CatBoost model using the parameters in Table 8.1, compare the results.

8.2.10 Experiment 10

In this experiment we wish to examine how prediction results change when we look at the wave at different times before the fault occurs with a fixed number of samples, and compare the results to Experiment 9.

Data

The wave form data-set shown in Tables 6.13, 6.14, 6.15, 6.17, 6.18, 6.19, 6.20, 6.21, 6.23, 6.24, 6.25, and 6.26, using combined aggregated values.

Method

For each data type, train a SVM, a random forest, a FFNN, and a CatBoost model using the parameters in Table 8.1, compare the results.

8.2.11 Experiment 11

In this experiment we wish to compare wavelet transform spectograms to wavelet scattering.

Data

The 1kHz wave form data-set shown in Table 6.1.

Method

Train a CNN for the spectograms, and a CatBoost model, a k-NN, a FFNN for the wavelet scattering using the parameters in Table 8.1, compare the results.

Chapter 9

Results

In this chapter we present and discuss the results from the experiments defined in Chapter 8.

9.1 Experiment 1

In Experiment 1 we examined whether aggregating the pure wave data gave comparable results to aggregating the Fourier coefficients and the RMS values of the wave, when looking at each second for one minute one minute before the faults occur. We compared non-faults to various faults, the results are presented in Table 9.1. Aside from ground faults, where Fourier coefficients achieved the best score, wave form performed best. Which model was the best alternated between CatBoost and random forest, depending on fault type. For general fault versus non-fault random forest appears to be the best suited model. We inspect the ROC curves for fault versus non-fault in Figures 9.1, 9.2, and 9.3. Wave form and RMS values appear to have an equally steep start, while Fourier coefficients do not. This indicates that aggregated values on the wave form and RMS values are able to identify a minority of the faults very well, while aggregated values on Fourier coefficients do not share this attribute. This means that even though RMS values reach a lower score than Fourier coefficients, it might be more suited for prediction, depending on what false positive rate is allowed. Even the best model have a high amount of incorrectly classified observations, as shown in Figure 9.4.

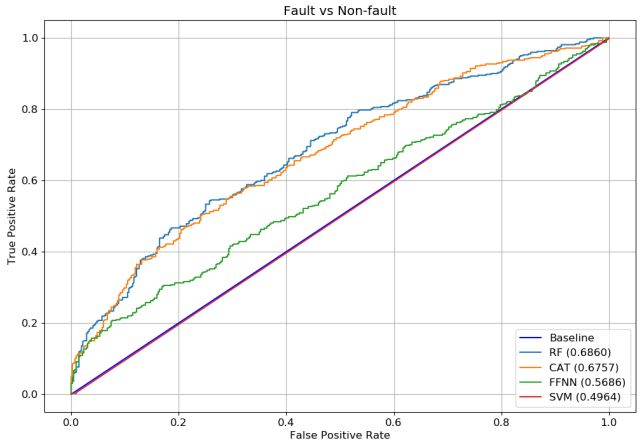


Figure 9.1: ROC curves for the combined aggregated values for faults versus non-faults for the 25kHz wave form data-set presented in Table 6.3.

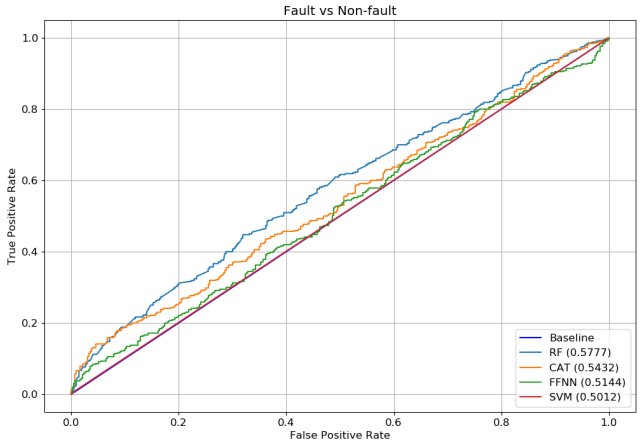


Figure 9.2: ROC curves for the combined aggregated values for faults versus non-faults for the 25kHz RMS value data-set presented in Table 6.5.

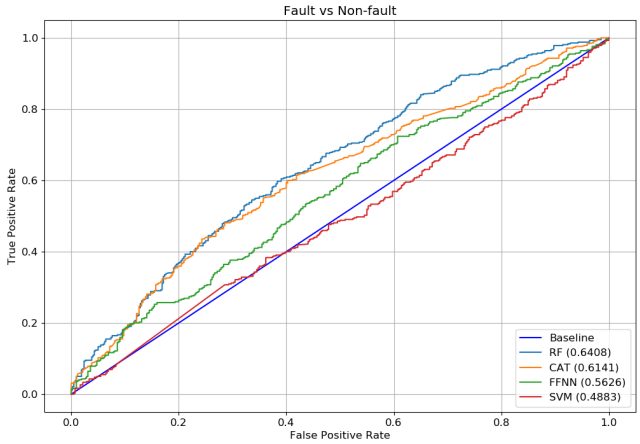


Figure 9.3: ROC curves for the combined aggregated values for faults versus non-faults for the 25kHz Fourier coefficient data-set presented in Table 6.6.

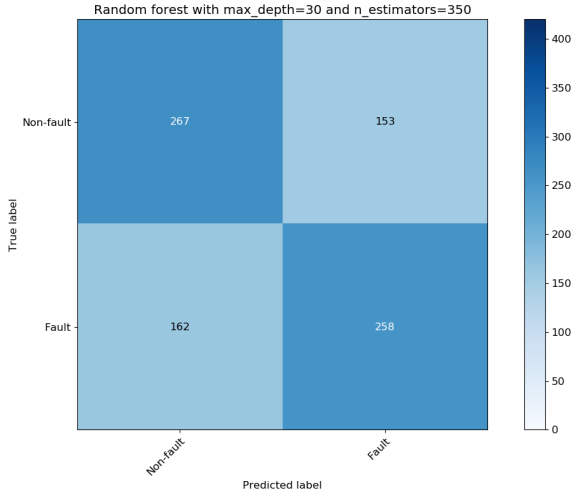


Figure 9.4: The confusion matrix for the combined aggregated values for faults versus non-faults for the 25kHz wave form data-set presented in Table 6.3.

9.2 Experiment 2

In Experiment 2 we examined whether aggregating the various data types using singular values gave comparable results to aggregating them using combined values. The results are presented in Table 9.2. Singular values scored convincingly lower than the combined values discussed in Experiment 1 for almost all data types and fault types. For singular values Fourier coefficients achieved the best score for fault versus non-fault, mostly due to the significant reduction in AUC-ROC score achieved with the wave form. Ground faults versus non-faults saw the only increase in score for the wave form, making this the highest scorer for that fault type thus far. It is reasonable to assume that the reduction in scores across the board is a consequence of overfitting, as singular values might contain too many features for the models to handle. Inspecting the training logs we find that the models get very high training accuracies while the validation- and test accuracies stay relatively low. This indicates that the models have the capacity to learn the data, which means that underfitting is not the issue. While the intent of preserving changes in time for the various aggregated values might be good, the implementation, or models used, are seemingly not.

9.3 Experiment 3

In Experiment 3 we examined whether looking at one second every minute for one hour before faults occurred gave similar results to looking at 60 consecutive seconds one minute before faults occurred. The results are presented in Tables 9.3 and 9.4 for combined and singular aggregated values respectively. For combined values Fourier coefficients achieve a similar score to the results in Experiment 1, while wave form and RMS score higher across the board, making wave form the best scorer for all comparisons. For singular values we see a similar trend, although this time Fourier also experience an increase in score. The increase is however not enough to offset the difference between combined and singular values already discussed in Experiment 2.

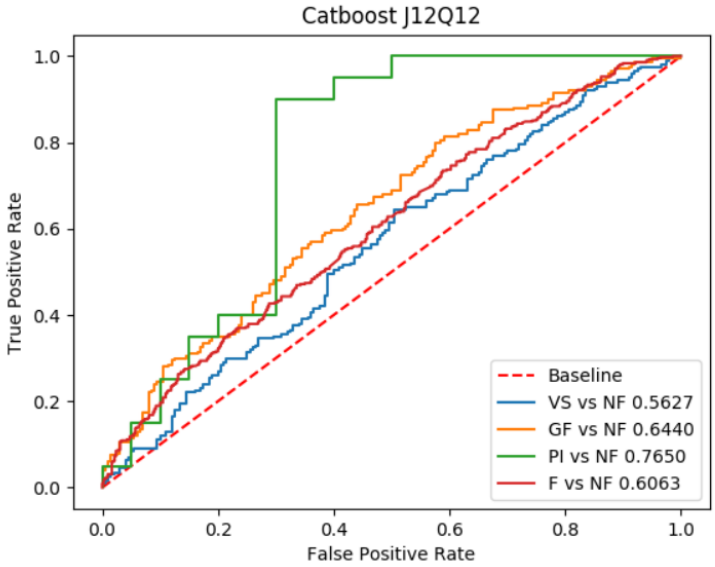
Wave form appears to be the data type best suited for predictions, despite the tendencies discussed in Chapter 7. Looking at the wave form one second every minute for one hour before the fault using combined values achieved the best score. That one second every minute for one hour before the fault achieved a better score than 60 seconds one minute before the fault suggests that there are indicators that a fault will happen hidden in the change of the wave for a long time prior to the fault, not just immediately before it, which are caught even when only looking at minor subsets over the period.

9.4 Experiment 4

In Experiment 4 we examined whether using wavelet scattering as feature extraction would give results comparable to those achieved by using aggregates values as features. The results are presented in Table 9.5. CatBoost achieved the best performance for all types of faults except ground faults where FFNN outperformed it. *k*-NN had the worst performance overall, not achieving a single best score for any fault type. Compared to the results from Experiments 1 and 2 presented in Tables 9.1 and 9.2 which used the same data-set, we can see that for the same data type – Wave form –, aggregated values outperform wavelet scattering significantly for all fault types. Comparing to the other data types and to the results presented in Tables 9.3 and 9.4 which used different data-sets, we can see that wavelet scattering is outperformed for all fault types.

We inspect the ROC curves for CatBoost in Figure 9.5. Comparing the general faults versus non-faults to the ROC curves plotted in Figures 9.1, 9.2 and 9.3, we can see that the the curve for wavelet scattering is not very steep, mostly resembling the curve in Figure 9.3. As mentioned in Experiment 1 this suggests that wavelet scattering with CatBoost is not very suited for prediction as even the top predictions are not too probable.

These results indicate that wavelet scattering is not as good as aggregation at capturing the features that characterize faults. It did however outperform aggregation when Fourier and RMS were used, but this was expected as the previous experiments have found wave form to achieve the best results. This might indicate that the faults are easier to identify by looking at simpler features than the more complex representation that the wavelet scattering creates. Wavelet scattering might capture some features that represent uninteresting noise or bias which are not captured by the simpler aggregation. It is also worth noting that the parameters for the wavelet scattering were not optimized as we only had time to test one pair of J and Q, meaning that other pairs might have yielded better results.



(a) Wave form

Figure 9.5: ROC curves for the wavelet scattering for the 25kHz wave form data-set presented in Table 6.3.

Freq	Data type	Features	Classifier	F vs NF	GF vs NF	VS vs NF	PI vs NF
25kHz	Wave form	Combined	CatBoost	0.6757 (0.6238)	**0.6534 (0.5900)**	**0.6292 (0.5750)**	0.7986 (0.7500)
			FFNN	0.5686 (0.5083)	0.6148 (0.6000)	0.5615 (0.5125)	0.7211 (0.6364)
			RF	**0.6860 (0.6250)** **MD=25,E=150**	0.6422 (0.6025) MD=15,E=50	0.6208 (0.5725) MD=None,E=50	0.7686 (0.7045) MD=5,E=250
			SVM	0.4964 (0.5190) K=sigmoid	0.5050 (0.4950) K=rbf	0.4947 (0.5000) K=sigmoid	**0.8017 (0.6818)** **K=linear**
25kHz	RMS values	Combined	CatBoost	0.5432 (0.5214)	**0.6365 (0.5900)**	0.5908 (0.5750)	**0.7541 (0.6591)**
			FFNN	0.5144 (0.5143)	0.5795 (0.5525)	0.6032 (0.5700)	0.5475 (0.5455)
			RF	**0.5777 (0.5548)** **MD=None,E=100**	0.6184 (0.5850) MD=20,E=500	**0.6121 (0.5750)** **MD=None,E=50**	0.6756 (0.6818) MD=15,E=150
			SVM	0.5012 (0.4964) K=sigmoid	0.4689 (0.4450) K=poly,D=6	0.5683 (0.5400) K=rbf	0.4329 (0.3864) K=poly,D=6
25kHz	Fourier coefficients	Combined	CatBoost	0.6141 (0.5929)	**0.7001 (0.6425)**	0.6071 (0.5800)	**0.7355 (0.7273)**
			FFNN	0.5626 (0.5369)	0.6272 (0.6075)	0.5082 (0.5150)	0.7087 (0.7273)
			RF	**0.6408 (0.6024)** **MD=15,E=350**	0.6773 (0.6100) MD=5,E=150	**0.6127 (0.5925)** **MD=25,E=300**	0.6942 (0.6818) MD=5,E=400
			SVM	0.4883 (0.5012) K=poly,D=2	0.4999 (0.4900) K=poly,D=2	0.4888 (0.5150) K=sigmoid	0.5537 (0.5455) K=rbf

Table 9.1: AUC-ROC scores for comparing balanced data-sets for various fault types using combined aggregated values on the 25kHz data-sets presented in Tables 6.3, 6.5, and 6.6. The best AUC-ROC score for each data-set is presented in bold, and the best score for each comparison across Tables 9.1 and 9.5 is highlighted in red, the accuracy is given in parentheses. Power interruptions versus non-faults is included for completeness, but due to the lack of instances of power interruptions, these scores ought to be considered an indicator more than a result.
The classifier parameters are abbreviated as: Max depth: MD, Estimators: E, Kernel: K, and Degree: D.
The fault types are abbreviated as: Faults: F, Non-faults: NF, Ground faults: GF, Voltage sags: VS, and Power interruptions: PI.

Freq	Data type	Features	Classifier	F vs NF	GF vs NF	VS vs NF	PI vs NF
25kHz	Wave form	Singular	CatBoost	0.5471 (0.5298)	0.6528 (0.5950)	**0.6105** (**0.5775**)	0.7624 (0.7045)
			FFNN	0.4720 (0.4810)	0.5096 (0.5025)	0.5705 (0.5275)	0.6384 (0.5227)
			RF	**0.5553** (**0.5417**) **MD=30,E=50**	**0.6822** (**0.6500**) **MD=30,E=100**	0.5515 (0.5400) MD=25,E=50	**0.7707** (**0.8182**) **MD=25,E=200**
			SVM	0.4930 (0.4964) K=linear	0.4748 (0.5600) K=rbf	0.5000 (0.5000) K=sigmoid	0.5351 (0.6136) K=poly,D=5
25kHz	RMS values	Singular	CatBoost	0.5532 (0.5286)	0.5752 (0.5750)	0.5111 (0.5075)	**0.5248** (**0.5682**)
			FFNN	0.5088 (0.5214)	0.5463 (0.5275)	0.4814 (0.4975)	0.5103 (0.4545)
			RF	**0.5678** (**0.5429**) **MD=15,E=50**	**0.5917** (**0.5775**) **MD=15,E=50**	**0.5712** (**0.5525**) **MD=25,E=450**	0.5103 (0.5227) MD=15,E=100
			SVM	0.5343 (0.5369) K=linear	0.4788 (0.5025) K=sigmoid	0.4875 (0.4975) K=rbf	0.5083 (0.5682) K=rbf
25kHz	Fourier coefficients	Singular	CatBoost	0.5662 (0.5548)	0.6415 (0.6050)	**0.5505** (**0.5350**)	0.6281 (0.5909)
			FFNN	0.4770 (0.5071)	0.5677 (0.5600)	0.5504 (0.5600)	0.4442 (0.5455)
			RF	**0.6092** (**0.5774**) **MD=25,E=350**	**0.6537** (**0.6000**) **MD=30,E=400**	0.5194 (0.5025) MD=20,E=50	**0.6395** (**0.6591**) **MD=10,E=300**
			SVM	0.5005 (0.5250) K=sigmoid	0.5344 (0.5050) K=rbf	0.4832 (0.5025) K=sigmoid	0.3781 (0.3409) K=poly,D=2

Table 9.2: AUC-ROC scores for comparing balanced data-sets for various fault types using singular aggregated values on the 25kHz data-sets presented in Tables 6.3, 6.5, and 6.6. The best AUC-ROC score for each data-set is presented in bold, and the best score for each comparison across Tables 9.1 and 9.5 is highlighted in red, the accuracy is given in parentheses. Power interruptions versus non-faults is included for completeness, but due to the lack of instances of power interruptions, these scores ought to be considered an indicator more than a result.
The classifier parameters are abbreviated as: Max depth: MD, Estimators: E, Kernel: K, and Degree: D.
The fault types are abbreviated as: Faults: F, Non-faults: NF, Ground faults: GF, Voltage sags: VS, and Power interruptions: PI.

Freq	Data type	Features	Classifier	F vs NF	GF vs NF	VS vs NF	PI vs NF
25kHz	Wave form	Combined	CatBoost	0.6743 (0.6179)	0.7001 (0.6475)	**0.6660 (0.6350)**	0.7457 (0.6977)
			FFNN	0.5742 (0.5631)	**0.7162 (0.6525)**	0.5495 (0.5400)	**0.7965 (0.7209)**
			RF	**0.7063 (0.6488)** MD=30,E=200	0.7154 (0.6575) MD=25,E=450	0.6491 (0.6075) MD=25,E=250	0.7446 (0.7209) MD=5,E=100
			SVM	0.5355 (0.5345) K=linear	0.5887 (0.5175) K=poly,D=2	0.5048 (0.4900) K=poly,D=2	0.7381 (0.6977) K=linear
25kHz	RMS values	Combined	CatBoost	0.5937 (0.5500)	**0.6443 (0.5925)**	0.6043 (0.5775)	0.6107 (0.6098)
			FFNN	0.5763 (0.5440)	0.5736 (0.5500)	0.5874 (0.5450)	0.5238 (0.5122)
			RF	**0.6182 (0.5762)** MD=None,E=150	0.6299 (0.5825) MD=15,E=50	**0.6339 (0.5875)** MD=20,E=100	**0.6952 (0.6098)** MD=None,E=100
			SVM	0.5379 (0.5286) K=rbf	0.5609 (0.4575) K=sigmoid	0.5738 (0.5750) K=poly,D=6	0.5226 (0.5854) K=poly,D=3
25kHz	Fourier coefficients	Combined	CatBoost	**0.6341 (0.5998)**	**0.7010 (0.6300)**	**0.5933 (0.5556)**	0.7048 (0.6585)
			FFNN	0.6044 (0.5663)	0.6549 (0.6150)	0.5713 (0.5480)	0.6619 (0.6341)
			RF	0.6317 (0.5986) MD=5,E=100	0.6869 (0.6175) MD=5,E=50	0.5904 (0.5783) MD=5,E=500	**0.7310 (0.7073)** MD=5,E=150
			SVM	0.5171 (0.5090) K=poly,D=6	0.6085 (0.5825) K=poly,D=6	0.5000 (0.5379) K=sigmoid	0.7238 (0.6829) K=poly,D=2

Table 9.3: AUC-ROC scores for comparing balanced data-sets for various fault types using combined aggregated values on the 25kHz data-sets presented in Tables 6.9, 6.11, and 6.12. The best AUC-ROC score for each data-set is presented in bold, and the best score for each comparison across Tables 9.1 and 9.5 is highlighted in red, the accuracy is given in parentheses. Power interruptions versus non-faults is included for completeness, but due to the lack of instances of power interruptions, these scores ought to be considered an indicator more than a result.
The classifier parameters are abbreviated as: Max depth: MD, Estimators: E, Kernel: K, and Degree: D.
The fault types are abbreviated as: Faults: F, Non-faults: NF, Ground faults: GF, Voltage sags: VS, and Power interruptions: PI.

Freq	Data type	Features	Classifier	F vs NF	GF vs NF	VS vs NF	PI vs NF
25kHz	Wave form	Singular	CatBoost	0.5711 (0.5524)	0.5864 (0.5950)	0.5469 (0.5325)	0.6667 (0.5814)
			FFNN	0.4966 (0.4917)	0.5239 (0.5125)	0.4824 (0.5025)	0.6320 (0.5349)
			RF	**0.6167** (**0.5738**) **MD=30,E=50**	**0.6897** (**0.6700**) **MD=25,E=350**	**0.5719** (**0.5350**) **MD=25,E=50**	**0.7879** (**0.7209**) **MD=15,E=150**
			SVM	0.4817 (0.5060) K=sigmoid	0.5000 (0.5050) K=sigmoid	0.4767 (0.4925) K=linear	0.2641 (0.6512) K=poly,D=5
25kHz	RMS values	Singular	CatBoost	**0.5696** (**0.5476**)	**0.5880** (**0.5500**)	0.5607 (0.5325)	0.6500 (0.6829)
			FFNN	0.5004 (0.5083)	0.5648 (0.5275)	0.5582 (0.5625)	0.4714 (0.4390)
			RF	0.5568 (0.5357) MD=20,E=100	0.5557 (0.5475) MD=25,E=50	0.5293 (0.5075) MD=15,E=50	**0.6702** (**0.6585**) **MD=15,E=300**
			SVM	0.5165 (0.5167) K=poly,D=3	0.5382 (0.5150) K=linear	**0.5728** (**0.5725**) **K=linear**	0.5774 (0.5366) K=linear
25kHz	Fourier coefficients	Singular	CatBoost	0.6225 (0.5878)	**0.6515** (**0.6150**)	**0.6040** (**0.5707**)	0.5405 (0.5366)
			FFNN	0.4930 (0.5006)	0.5316 (0.5225)	0.4488 (0.4798)	0.4238 (0.3902)
			RF	**0.6468** (**0.6093**) **MD=25,E=150**	0.6455 (0.6050) MD=5,E=150	0.5775 (0.5556) MD=15,E=50	**0.7190** (**0.6098**) **MD=5,E=50**
			SVM	0.5076 (0.4982) K=sigmoid	0.5439 (0.5400) K=rbf	0.4970 (0.5152) K=linear	0.4619 (0.5366) K=rbf

Table 9.4: AUC-ROC scores for comparing balanced data-sets for various fault types using singular aggregated values on the 25kHz data-sets presented in Tables 6.9, 6.11, and 6.12. The best AUC-ROC score for each data-set is presented in bold, and the best score for each comparison across Tables 9.1 and 9.5 is highlighted in red, the accuracy is given in parentheses. Power interruptions versus non-faults is included for completeness, but due to the lack of instances of power interruptions, these scores ought to be considered an indicator more than a result.
The classifier parameters are abbreviated as: Max depth: MD, Estimators: E, Kernel: K, and Degree: D.
The fault types are abbreviated as: Faults: F, Non-faults: NF, Ground faults: GF, Voltage sags: VS, and Power interruptions: PI.

Freq	Data type	Features	Classifier	F vs NF	GF vs NF	VS vs NF	PI vs NF
25kHz	Wave form	Wavelet scattering J=12,Q=12	CatBoost	**0.6063** (**0.5641**)	0.6440 (0.6025)	**0.5627** (**0.5525**)	**0.7650** (**0.8000**)
			FFNN	0.5220 (0.5310)	**0.6932** (**0.6325**)	0.5210 (0.5100)	0.7526 (0.6500)
			k-NN	0.5191 (0.4989)	0.5995 (0.5871)	0.5551 (0.5525)	0.7002 (0.6667)

Table 9.5: AUC-ROC scores for comparing balanced data-sets for various fault types using wavelet scattering on the 25kHz wave form data-set presented in Table 6.3. The best AUC-ROC score for each data-set is presented in bold, and the best score for each comparison across Tables 9.1 and 9.5 is highlighted in red, the accuracy is given in parentheses. Power interruptions versus non-faults is included for completeness, but due to the lack of instances of power interruptions, these scores ought to be considered an indicator more than a result.
The fault types are abbreviated as: Faults: F, Non-faults: NF, Ground faults: GF, Voltage sags: VS, and Power interruptions: PI.

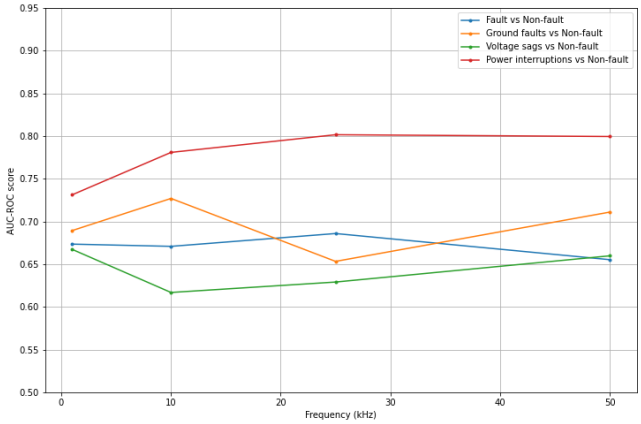


Figure 9.6: The best AUC-ROC scores for different frequencies for Table 9.6.

9.5 Experiment 5

In Experiment 5 we examined whether aggregating the pure wave data using combined values on different frequencies gave notable differences in predictability. The results are presented in Table 9.6, and the best scores for each frequency are presented in Figure 9.6. While there are significant variances in the scores between the different frequencies, we see no clear pattern of scores neither increasing nor decreasing as frequency increases.

9.6 Experiment 6

In Experiment 6 we examined whether aggregating the pure wave data using singular values on different frequencies gave notable differences in predictability, and differed from the results when using combined aggregated values. The results are presented in Table 9.7, and the best scores for each frequency are presented in Figure 9.7. In contrast to the results from Experiment 5 we see a upwards trend in scores as frequency increases. This likely stems from singular values aggregating over smaller time windows, which allows noise to have a noticeable impact on the calculated values, while combined values considers too many points, which makes the noise vanish. Despite the upwards trend combined values still have better scores than singular values, for the same reason as that discussed in Experiment 2.

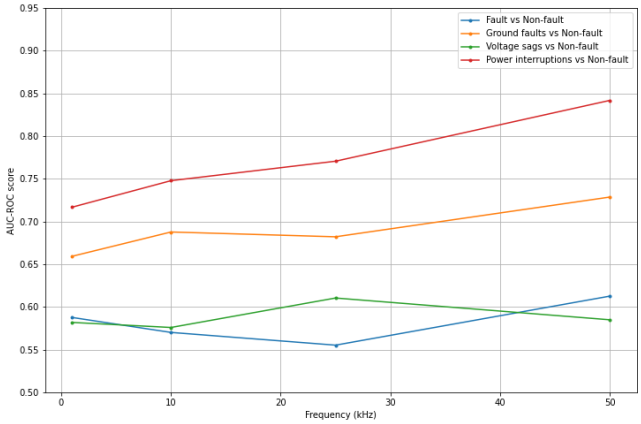


Figure 9.7: The best AUC-ROC scores for different frequencies for Table 9.7.

9.7 Experiment 7

In Experiment 7 we examined whether looking at one second every minute for one hour before faults occurred gave similar results to looking at 60 consecutive seconds a minute before faults occurred for different frequencies. The results are presented in Tables 9.8 and 9.9, and the best scores are presented in Figures 9.8 and 9.9. Again we see no clear pattern of scores neither increasing nor decreasing as frequency increases, even for singular values, which showed such a tendency in Experiment 6. This might suggest that noise increases noticeably shortly prior to a fault, but not to an impactful degree before that. While there is some variance in the AUC-ROC scores, there is no consistency in which duration scores best. We see no indication that looking at 60 seconds one minute before the fault gives neither better nor worse predictability than looking at one second every minute for one hour before the fault, but that one minute one minute before the fault contains more information which is better utilized at higher frequencies.

9.8 Experiment 8

In Experiment 8 we examined whether using wavelet scattering on different frequencies gave notable differences in predictability. The results are presented in Table 9.10. CatBoost achieved the overall best performance for both frequencies, with FFNN outperforming it on Ground faults for 1kHz and 25kHz, and on power interruptions for 1kHz. *k*-NN had the worst performance overall, not achieving a single best score for any fault type. Compared to the results from Experiments 5, 6 and 7 presented in Tables 9.6, 9.7, 9.8 and

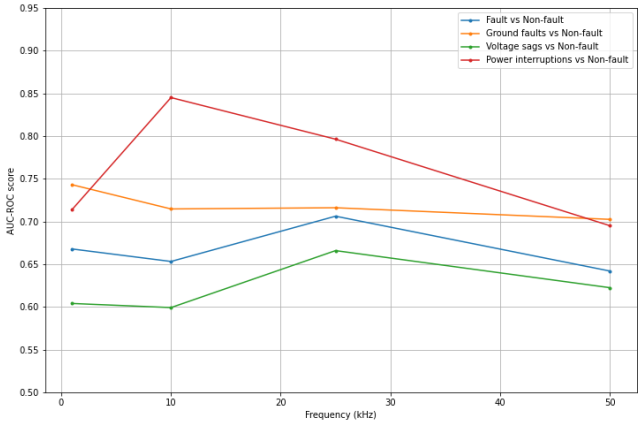


Figure 9.8: The best AUC-ROC scores for different frequencies for Table 9.8.

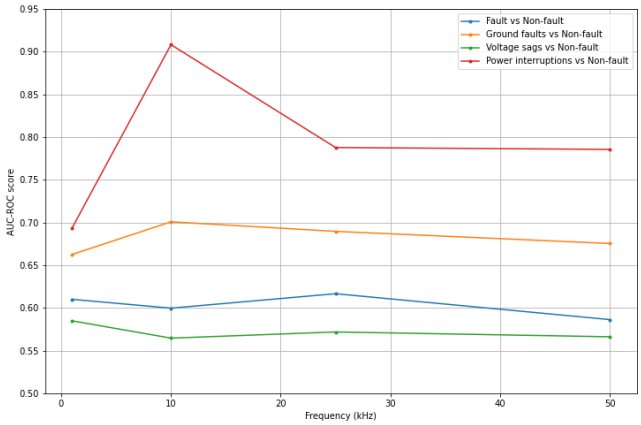


Figure 9.9: The best AUC-ROC scores for different frequencies for Table 9.9.

9.9, we can see that also for wavelet scattering there is not a frequency that dominates the other in terms of performance. A higher frequency does not seem to either improve nor worsen the performance.

That the sampled frequency did not have any significant impact on the results suggests that there is not much useful information to gain from looking at minor static or high frequency noise, but that the fault rather ought to be detected by looking at changes in the characteristics of the wave over a longer period of time.

Freq	Data type	Features	Classifier	F vs NF	GF vs NF	VS vs NF	PI vs NF
1kHz	Wave form	Combined	CatBoost	0.6675 (0.6310)	**0.6895** (**0.6275**)	**0.6674** (**0.6150**)	0.6942 (0.6818)
			FFNN	0.5385 (0.5202)	0.5922 (0.5825)	0.4974 (0.5050)	0.6467 (0.6364)
			RF	**0.6736** (**0.6405**) **MD=30,E=300**	0.6713 (0.6225) MD=None,E=150	0.6516 (0.6275) MD=25,E=100	0.6952 (0.6591) MD=25,E=100
			SVM	0.5795 (0.5548) K=linear	0.4546 (0.5000) K=rbf	0.5060 (0.5175) K=sigmoid	**0.7314** (**0.7045**) **K=poly,D=6**
10kHz	Wave form	Combined	CatBoost	0.6477 (0.6226)	**0.7271** (**0.6575**)	0.6072 (0.6000)	0.7686 (0.6591)
			FFNN	0.5702 (0.5583)	0.5722 (0.5650)	0.5018 (0.5125)	0.6033 (0.5909)
			RF	**0.6710** (**0.6274**) **MD=30,E=100**	0.7113 (0.6425) MD=None,E=50	**0.6170** (**0.6175**) **MD=15,E=50**	**0.7810** (**0.7045**) **MD=5,E=200**
			SVM	0.5000 (0.4988) K=sigmoid	0.4392 (0.5350) K=rbf	0.4837 (0.5000) K=sigmoid	0.7603 (0.7273) K=linear
25kHz	Wave form	Combined	CatBoost	0.6757 (0.6238)	**0.6534** (**0.5900**)	**0.6292** (**0.5750**)	0.7986 (0.7500)
			FFNN	0.5686 (0.5083)	0.6148 (0.6000)	0.5615 (0.5125)	0.7211 (0.6364)
			RF	**0.6860** (**0.6250**) **MD=25,E=150**	0.6422 (0.6025) MD=15,E=50	0.6208 (0.5725) MD=None,E=50	0.7686 (0.7045) MD=5,E=250
			SVM	0.4964 (0.5190) K=sigmoid	0.5050 (0.4950) K=rbf	0.4947 (0.5000) K=sigmoid	**0.8017** (**0.6818**) **K=linear**
50kHz	Wave form	Combined	CatBoost	0.6249 (0.5940)	**0.7111** (**0.6500**)	0.6172 (0.5725)	0.7397 (0.6818)
			FFNN	0.5334 (0.5262)	0.5195 (0.5150)	0.4970 (0.5075)	0.5145 (0.5455)
			RF	**0.6554** (**0.6155**) **MD=None,E=50**	0.6938 (0.6325) MD=5,E=50	**0.6599** (**0.6200**) **MD=30,E=250**	0.7355 (0.7045) MD=10,E=50
			SVM	0.5312 (0.5238) K=linear	0.5025 (0.6025) K=linear	0.5001 (0.4975) K=sigmoid	**0.7996** (**0.7727**) **K=linear**

Table 9.6: AUC-ROC scores for comparing balanced data-sets for various fault types using combined aggregated values on the wave form data-sets presented in Tables 6.1, 6.2, 6.3, and 6.4. The best AUC-ROC score for each data-set is presented in bold, and the best score for each comparison across Tables 9.6 and 9.10 is highlighted in red, the accuracy is given in parentheses. Power interruptions versus non-faults is included for completeness, but due to the lack of instances of power interruptions, these scores ought to be considered an indicator more than a result.
The classifier parameters are abbreviated as: Max depth: MD, Estimators: E, Kernel: K, and Degree: D.
The fault types are abbreviated as: Faults: F, Non-faults: NF, Ground faults: GF, Voltage sags: VS, and Power interruptions: PI.

Freq	Data type	Features	Classifier	F vs NF	GF vs NF	VS vs NF	PI vs NF
1kHz	Wave form	Singular	CatBoost	0.5552 (0.5452)	**0.6594** (**0.6175**)	0.5738 (0.5650)	**0.7169** (**0.6364**)
			FFNN	0.5173 (0.5060)	0.5615 (0.5400)	0.5126 (0.5100)	0.5909 (0.5455)
			RF	**0.5877** (**0.5726**) **MD=20,E=50**	0.6493 (0.6175) MD=30,E=50	**0.5819** (**0.5525**) **MD=25,E=50**	0.6188 (0.5455) MD=15,E=250
			SVM	0.5000 (0.5024) K=linear	0.5527 (0.5625) K=poly,D=2	0.4819 (0.4800) K=linear	0.5702 (0.5000) K=linear
10kHz	Wave form	Singular	CatBoost	0.5453 (0.5333)	0.6774 (0.6250)	0.5197 (0.5250)	0.6839 (0.6591)
			FFNN	0.5165 (0.4976)	0.4519 (0.4625)	0.4821 (0.5125)	0.6798 (0.5682)
			RF	**0.5702** (**0.5440**) **MD=25,E=150**	**0.6878** (**0.6400**) **MD=30,E=250**	**0.5760** (**0.5525**) **MD=None,E=50**	**0.7479** (**0.6136**) **MD=10,E=50**
			SVM	0.5036 (0.5131) K=linear	0.5342 (0.4700) K=rbf	0.5133 (0.4975) K=sigmoid	0.3781 (0.6364) K=linear
25kHz	Wave form	Singular	CatBoost	0.5471 (0.5298)	0.6528 (0.5950)	**0.6105** (**0.5775**)	0.7624 (0.7045)
			FFNN	0.4720 (0.4810)	0.5096 (0.5025)	0.5705 (0.5275)	0.6384 (0.5227)
			RF	**0.5553** (**0.5417**) **MD=30,E=50**	**0.6822** (**0.6500**) **MD=30,E=100**	0.5515 (0.5400) MD=25,E=50	**0.7707** (**0.8182**) **MD=25,E=200**
			SVM	0.4930 (0.4964) K=linear	0.4748 (0.5600) K=rbf	0.5000 (0.5000) K=sigmoid	0.5351 (0.6136) K=poly,D=5
50kHz	Wave form	Singular	CatBoost	**0.6127** (**0.5845**)	0.7256 (0.6625)	**0.5850** (**0.5950**)	0.8326 (0.8182)
			FFNN	0.4979 (0.5071)	0.5734 (0.5100)	0.4723 (0.4850)	0.4752 (0.4773)
			RF	0.5869 (0.5548) MD=30,E=50	**0.7287** (**0.6600**) **MD=25,E=150**	0.5791 (0.5550) MD=15,E=50	**0.8419** (**0.7727**) **MD=30,E=350**
			SVM	0.5000 (0.5083) K=rbf	0.5058 (0.4975) K=linear	0.5126 (0.5075) K=poly,D=2	0.5919 (0.6136) K=linear

Table 9.7: AUC-ROC scores for comparing balanced data-sets for various fault types using singular aggregated values on the wave form data-sets presented in Tables 6.1, 6.2, 6.3, and 6.4. The best AUC-ROC score for each data-set is presented in bold, and the best score for each comparison across Tables 9.6 and 9.10 is highlighted in red, the accuracy is given in parentheses. Power interruptions versus non-faults is included for completeness, but due to the lack of instances of power interruptions, these scores ought to be considered an indicator more than a result.
The classifier parameters are abbreviated as: Max depth: MD, Estimators: E, Kernel: K, and Degree: D.
The fault types are abbreviated as: Faults: F, Non-faults: NF, Ground faults: GF, Voltage sags: VS, and Power interruptions: PI.

Freq	Data type	Features	Classifier	F vs NF	GF vs NF	VS vs NF	PI vs NF
1kHz	Wave form	Combined	CatBoost	**0.6679 (0.6321)**	**0.7431 (0.6725)**	**0.6041 (0.5700)**	**0.7143 (0.6341)**
			FFNN	0.5803 (0.5202)	0.6943 (0.6775)	0.5354 (0.5250)	0.5238 (0.4634)
			RF	0.6611 (0.6238) MD=20,E=50	0.7104 (0.6650) MD=5,E=50	0.5981 (0.5550) MD=15,E=100	0.6405 (0.5854) MD=5,E=100
			SVM	0.5107 (0.5131) K=poly,D=2	0.6581 (0.6175) K=linear	0.4954 (0.4950) K=poly,D=4	0.6571 (0.6341) K=linear
10kHz	Wave form	Combined	CatBoost	0.6454 (0.6060)	**0.7148 (0.6525)**	0.5758 (0.5500)	0.7905 (0.7317)
			FFNN	0.5709 (0.5619)	0.6298 (0.5400)	0.5017 (0.5275)	0.5738 (0.4878)
			RF	**0.6533 (0.6298)** **MD=None,E=200**	0.7147 (0.6650) MD=None,E=50	**0.5993 (0.5925)** **MD=None,E=200**	**0.8452 (0.7073)** **MD=None,E=300**
			SVM	0.4931 (0.5071) K=rbf	0.5000 (0.4875) K=linear	0.5060 (0.4875) K=poly,D=3	0.7524 (0.7073) K=linear
25kHz	Wave form	Combined	CatBoost	0.6743 (0.6179)	0.7001 (0.6475)	**0.6660 (0.6350)**	0.7457 (0.6977)
			FFNN	0.5742 (0.5631)	**0.7162 (0.6525)**	0.5495 (0.5400)	**0.7965 (0.7209)**
			RF	**0.7063 (0.6488)** **MD=30,E=200**	0.7154 (0.6575) MD=25,E=450	0.6491 (0.6075) MD=25,E=250	0.7446 (0.7209) MD=5,E=100
			SVM	0.5355 (0.5345) K=linear	0.5887 (0.5175) K=poly,D=2	0.5048 (0.4900) K=poly,D=2	0.7381 (0.6977) K=linear
50kHz	Wave form	Combined	CatBoost	**0.6421 (0.6095)**	0.6727 (0.6450)	0.6190 (0.5850)	**0.6952 (0.6829)**
			FFNN	0.5167 (0.5262)	0.5324 (0.5325)	0.4836 (0.5000)	0.5881 (0.5366)
			RF	0.6319 (0.6071) MD=None,E=50	**0.7026 (0.6400)** **MD=20,E=150**	**0.6226 (0.5775)** **MD=30,E=200**	0.6905 (0.6585) MD=5,E=100
			SVM	0.5266 (0.4952) K=rbf	0.4282 (0.5150) K=rbf	0.4440 (0.5075) K=rbf	0.6286 (0.5610) K=poly,D=6

Table 9.8: AUC-ROC scores for comparing balanced data-sets for various fault types using combined aggregated values on the wave form data-sets presented in Tables 6.7, 6.8, 6.9, and 6.10. The best AUC-ROC score for each data-set is presented in bold, and the best score for each comparison across Tables 9.6 and 9.10 is highlighted in red, the accuracy is given in parentheses. Power interruptions versus non-faults is included for completeness, but due to the lack of instances of power interruptions, these scores ought to be considered an indicator more than a result.
The classifier parameters are abbreviated as: Max depth: MD, Estimators: E, Kernel: K, and Degree: D.
The fault types are abbreviated as: Faults: F, Non-faults: NF, Ground faults: GF, Voltage sags: VS, and Power interruptions: PI.

Freq	Data type	Features	Classifier	F vs NF	GF vs NF	VS vs NF	PI vs NF
1kHz	Wave form	Singular	CatBoost	0.5692 (0.5345)	0.6499 (0.5950)	0.5685 (0.5325)	0.6000 (0.5854)
			FFNN	0.5000 (0.5012)	0.5309 (0.5325)	0.5219 (0.5125)	**0.6940 (0.6341)**
			RF	**0.6101 (0.5726)** **MD=25,E=350**	**0.6627 (0.6050)** **MD=25,E=100**	**0.5849 (0.5525)** **MD=25,E=100**	0.6702 (0.6829) MD=None,E=100
			SVM	0.5102 (0.5012) K=linear	0.5447 (0.5125) K=linear	0.5437 (0.4975) K=sigmoid	0.6679 (0.6585) K=linear
10kHz	Wave form	Singular	CatBoost	0.5949 (0.5845)	0.6436 (0.5900)	**0.5647 (0.5375)**	0.8976 (0.8049)
			FFNN	0.4964 (0.5036)	0.5699 (0.5500)	0.4999 (0.5100)	0.7429 (0.6341)
			RF	**0.5998 (0.5595)** **MD=30,E=50**	**0.7009 (0.6375)** **MD=10,E=400**	0.5466 (0.5225) MD=15,E=50	**0.9083 (0.8049)** **MD=25,E=50**
			SVM	0.4980 (0.4952) K=sigmoid	0.5371 (0.4950) K=rbf	0.5000 (0.5000) K=linear	0.6690 (0.7317) K=linear
25kHz	Wave form	Singular	CatBoost	0.5711 (0.5524)	0.5864 (0.5950)	0.5469 (0.5325)	0.6667 (0.5814)
			FFNN	0.4966 (0.4917)	0.5239 (0.5125)	0.4824 (0.5025)	0.6320 (0.5349)
			RF	**0.6167 (0.5738)** **MD=30,E=50**	**0.6897 (0.6700)** **MD=25,E=350**	**0.5719 (0.5350)** **MD=25,E=50**	**0.7879 (0.7209)** **MD=15,E=150**
			SVM	0.4817 (0.5060) K=sigmoid	0.5000 (0.5050) K=sigmoid	0.4767 (0.4925) K=linear	0.2641 (0.6512) K=poly,D=5
50kHz	Wave form	Singular	CatBoost	0.5325 (0.5214)	0.6419 (0.6175)	0.5617 (0.5325)	0.7357 (0.7073)
			FFNN	0.4725 (0.4988)	0.5558 (0.4975)	0.4865 (0.5150)	0.6857 (0.7073)
			RF	**0.5864 (0.5595)** **MD=None,E=50**	**0.6755 (0.6450)** **MD=30,E=50**	**0.5663 (0.5300)** **MD=15,E=50**	**0.7857 (0.7317)** **MD=None,E=400**
			SVM	0.4931 (0.4988) K=linear	0.5164 (0.4875) K=linear	0.5125 (0.4900) K=poly,D=2	0.6762 (0.6829) K=linear

Table 9.9: AUC-ROC scores for comparing balanced data-sets for various fault types using singular aggregated values on the wave form data-sets presented in Tables 6.7, 6.8, 6.9, and 6.10. The best AUC-ROC score for each data-set is presented in bold, and the best score for each comparison across Tables 9.6 and 9.10 is highlighted in red, the accuracy is given in parentheses. Power interruptions versus non-faults is included for completeness, but due to the lack of instances of power interruptions, these scores ought to be considered an indicator more than a result.
The classifier parameters are abbreviated as: Max depth: MD, Estimators: E, Kernel: K, and Degree: D.
The fault types are abbreviated as: Faults: F, Non-faults: NF, Ground faults: GF, Voltage sags: VS, and Power interruptions: PI.

Freq	Data type	Features	Classifier	F vs NF	GF vs NF	VS vs NF	PI vs NF
1kHz	Wave form	Wavelet scattering	CatBoost	**0.5855** (**0.5570**) **J=15,Q=12**	0.6652 (0.6275) J=15,Q=24	**0.5948** (**0.6025**) **J=12,Q=96**	0.7675 (0.7250) J=6,Q=48
			FFNN	0.5230 (0.4917) J=6,Q=12	**0.6798** (**0.6275**) **J=8,Q=12**	0.5505 (0.5575) J=8,Q=12	**0.7734 (0.7000) J=15,Q=12**
			k-NN	0.5508 (0.5357) J=12,Q=12	0.6101 (0.5676) J=8,Q=12	0.5604 (0.5691) J=12,Q=12	0.7057 (0.6667) J=12,Q=24
25kHz	Wave form	Wavelet scattering J=12,Q=12	CatBoost	**0.6063** (**0.5641**)	0.6440 (0.6025)	**0.5627** (**0.5525**)	**0.7650** (**0.8000**)
			FFNN	0.5220 (0.5310)	**0.6932** (**0.6325**)	0.5210 (0.5100)	0.7526 (0.6500)
			k-NN	0.5191 (0.4989)	0.5995 (0.5871)	0.5551 (0.5525)	0.7002 (0.6667)

Table 9.10: AUC-ROC scores for comparing balanced data-sets for various fault types using wavelet scattering on the wave form data-sets presented in Tables 6.1 and 6.3. The best AUC-ROC score for each data-set is presented in bold, and the best score for each comparison across Tables 9.6 and 9.10 is highlighted in red, the accuracy is given in parentheses. Power interruptions versus non-faults is included for completeness, but due to the lack of instances of power interruptions, these scores ought to be considered an indicator more than a result.
The fault types are abbreviated as: Faults: F, Non-faults: NF, Ground faults: GF, Voltage sags: VS, and Power interruptions: PI.

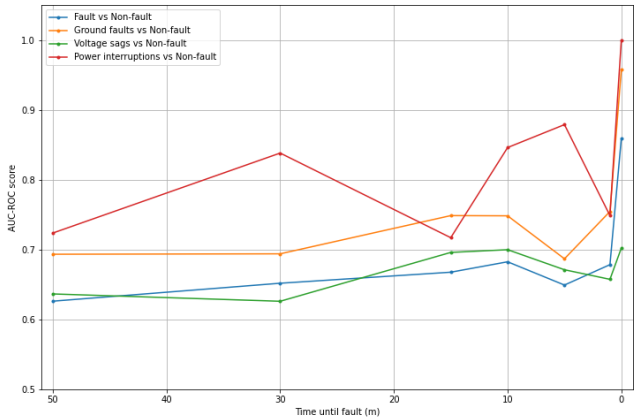


Figure 9.10: The best AUC-ROC scores for different times until fault for Table 9.11 and 9.12.

9.9 Experiment 9

In Experiment 9 we examined how looking at different times before the fault occurred affected our results. The results are presented in Tables 9.11 and 9.12, and the best scores are presented in Figure 9.10. For most faults the difference between removing one minute before the fault occurs, and 50 minutes before the fault occurs, is quite small. It appears that looking at data up until 10 to 15 minutes before the fault occurs gives some of the better results. Ground faults and power interruptions appear to be very differentiable from non-faults when looking at 0 minutes before the fault, while the difference between 1 minute and 50 minutes before the fault is comparably small, as was suggested in Section 7.3.

There is a sudden drop in the AUC-ROC for power interruptions at 1 minute before the fault occurs. This seems a bit counter-intuitive as more data should not worsen the result by such a big amount. To see what might cause the drops we look at the features the classifiers are using for the different forecast horizons. The V1 max aggregation for data sampled over one second each minute an hour leading up to the faults is shown in Figure 9.11. We can see that the 95th percentile for power interruptions is clearly separated from the other faults until around 5 minutes before the fault occurs, where it suddenly jumps up to the same values as the other faults. This makes it harder to differentiate when looking at this feature and explains why it is harder to classify it when this time interval is included. As the features used are aggregations which simplifies the data, looking at a too big time interval might worsen the performance as seen here, as some features that are easy to distinguish at some time intervals might get overwritten by some that are less differentiating.

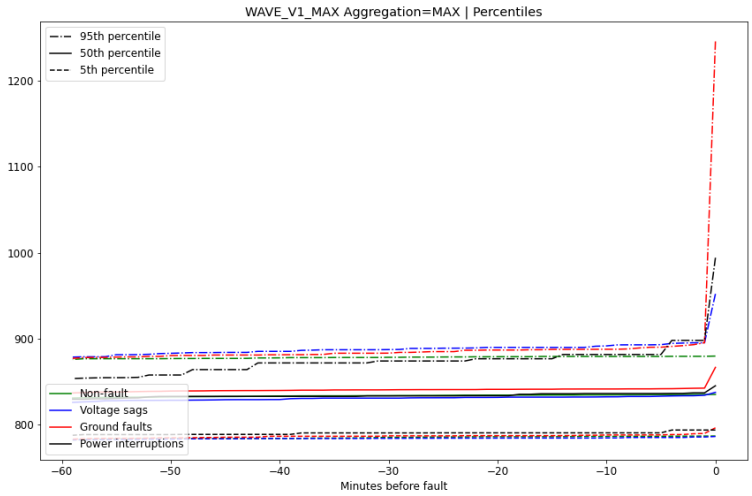


Figure 9.11: The V1 max aggregation using the combined aggregated values on the 1kHz wave form data-sets presented in Tables 6.13, 6.14, 6.15, 6.17, 6.18, and 6.19. The data is aggregated using the maximum function, and at x minutes before the fault the data is aggregated from 60 minutes before the fault up to x minutes before the fault, resulting in the same feature that the classifiers are using.

Time until fault	Data duration	Classifier	F vs NF	GF vs NF	VS vs NF	PI vs NF
0 min	60 min	CatBoost	0.8577 (0.7914)	**0.9576 (0.8950)**	0.6786 (0.6170)	0.9947 (0.9487)
		FFNN	**0.8593 (0.8098)**	0.9447 (0.8875)	0.5634 (0.5346)	**1.0000 (0.9744)**
		RF	0.8357 (0.7840) MD=5,E=500	0.9489 (0.8950) MD=10,E=200	**0.7022 (0.6516) MD=25,E=500**	1.0000 (1.0000) MD=10,E=50
		SVM	0.8289 (0.7239) K=poly,D=6	0.9134 (0.8900) K=linear	0.6610 (0.6303) K=linear	0.9921 (0.9231) K=linear
1 min	59 min	CatBoost	0.6718 (0.6221)	0.7462 (0.6675)	0.6330 (0.6011)	0.7263 (0.6410)
		FFNN	0.5513 (0.5227)	0.6125 (0.5825)	0.5524 (0.5824)	0.5737 (0.5385)
		RF	**0.6785 (0.6196) MD=None,E=150**	**0.7546 (0.6850) MD=10,E=50**	**0.6575 (0.6170) MD=25,E=50**	**0.7487 (0.6667) MD=10,E=500**
		SVM	0.5455 (0.5301) K=rbf	0.4729 (0.5150) K=rbf	0.6260 (0.5638) K=linear	0.6934 (0.6410) K=linear
5 min	55 min	CatBoost	0.6333 (0.5939)	0.6795 (0.6125)	0.6164 (0.5585)	0.8553 (0.7949)
		FFNN	0.5755 (0.5558)	0.5680 (0.5250)	0.5833 (0.5585)	0.6868 (0.5897)
		RF	**0.6493 (0.6110) MD=25,E=450**	**0.6869 (0.6475) MD=25,E=250**	**0.6712 (0.6250) MD=None,E=100**	**0.8789 (0.7692) MD=15,E=50**
		SVM	0.5196 (0.5178) K=poly,D=2	0.4072 (0.5025) K=linear	0.5000 (0.5080) K=linear	0.7316 (0.6667) K=linear
10 min	50 min	CatBoost	0.6597 (0.6147)	**0.7484 (0.6850)**	**0.6999 (0.6489)**	0.8105 (0.7179)
		FFNN	0.4888 (0.5043)	0.6110 (0.5600)	0.6236 (0.5665)	0.5474 (0.5128)
		RF	**0.6826 (0.6331) MD=25,E=150**	0.7473 (0.6800) MD=15,E=150	0.6960 (0.6676) MD=25,E=400	**0.8461 (0.7179) MD=5,E=50**
		SVM	0.4986 (0.5264) K=linear	0.6154 (0.5900) K=rbf	0.5000 (0.5000) K=sigmoid	0.7592 (0.6923) K=poly,D=6

Table 9.11: AUC-ROC scores for comparing balanced data-sets for various fault types using combined aggregated values on the 1kHz wave form data-sets presented in Tables 6.13, 6.14, 6.15, 6.17, 6.18, and 6.19. The best AUC-ROC score for each data-set is presented in bold, and the best score for each comparison across Tables 9.11 and 9.12 is highlighted in red, the accuracy is given in parentheses. Power interruptions versus non-faults is included for completeness, but due to the lack of instances of power interruptions, these scores ought to be considered an indicator more than a result.
The classifier parameters are abbreviated as: Max depth: MD, Estimators: E, Kernel: K, and Degree: D.
The fault types are abbreviated as: Faults: F, Non-faults: NF, Ground faults: GF, Voltage sags: VS, and Power interruptions: PI. 1/2

Time until fault	Data duration	Classifier	F vs NF	GF vs NF	VS vs NF	PI vs NF
15 min	45 min	CatBoost	0.6607 (0.6135)	0.7473 (0.6725)	0.6764 (0.6170)	0.7000 (0.6410)
		FFNN	0.5643 (0.5595)	0.5086 (0.5250)	0.5773 (0.5505)	0.6658 (0.6410)
		RF	**0.6677** (**0.6061**) **MD=None,E=450**	**0.7487** (**0.6850**) **MD=10,E=50**	**0.6960** (**0.6569**) **MD=None,E=200**	**0.7171** (**0.6410**) **MD=5,E=50**
		SVM	0.4722 (0.5055) K=linear	0.3816 (0.5150) K=rbf	0.5261 (0.5053) K=poly,D=6	0.6474 (0.6410) K=linear
30 min	30 min	CatBoost	0.6318 (0.5951)	0.6925 (0.6525)	**0.6261** (**0.6090**)	0.7658 (0.6923)
		FFNN	0.5691 (0.5387)	0.5181 (0.5600)	0.5716 (0.5186)	0.6000 (0.5128)
		RF	**0.6519** (**0.6098**) **MD=None,E=50**	**0.6940** (**0.6550**) **MD=10,E=50**	0.6067 (0.5851) MD=20,E=250	0.7855 (0.6923) MD=15,E=50
		SVM	0.5005 (0.5055) K=linear	0.4517 (0.5100) K=rbf	0.4559 (0.5160) K=sigmoid	**0.8382** (**0.8205**) **K=linear**
50 min	10 min	CatBoost	**0.6262** (**0.6025**)	0.6860 (0.6350)	0.6303 (0.5878)	0.6987 (0.7179)
		FFNN	0.5400 (0.5313)	0.5199 (0.5200)	0.4785 (0.4894)	0.6447 (0.6154)
		RF	0.6221 (0.5890) MD=20,E=150	**0.6934** (**0.6525**) **MD=30,E=50**	**0.6366** (**0.5878**) **MD=20,E=50**	**0.7237** (**0.7436**) **MD=25,E=100**
		SVM	0.5000 (0.4994) K=poly,D=2	0.6642 (0.6025) K=rbf	0.4776 (0.5532) K=poly,D=2	0.7211 (0.6667) K=linear

Table 9.12: AUC-ROC scores for comparing balanced data-sets for various fault types using combined aggregated values on the 1kHz wave form data-sets presented in Tables 6.13, 6.14, 6.15, 6.17, 6.18, and 6.19. The best AUC-ROC score for each data-set is presented in bold, and the best score for each comparison across Tables 9.11 and 9.12 is highlighted in red, the accuracy is given in parentheses. Power interruptions versus non-faults is included for completeness, but due to the lack of instances of power interruptions, these scores ought to be considered an indicator more than a result.
The classifier parameters are abbreviated as: Max depth: MD, Estimators: E, Kernel: K, and Degree: D.
The fault types are abbreviated as: Faults: F, Non-faults: NF, Ground faults: GF, Voltage sags: VS, and Power interruptions: PI. 2/2

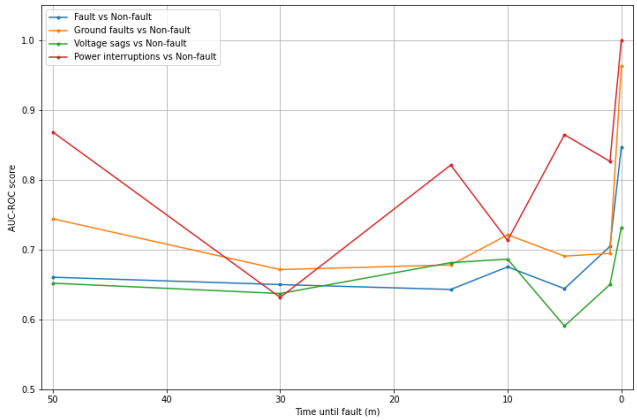


Figure 9.12: The best AUC-ROC scores for different times until fault for Table 9.13 and 9.14.

9.10 Experiment 10

In Experiment 10 we examined how looking at different times before the fault occurred affected our results when we have a fixed number of samples per observation. The results are presented in Tables 9.13 and 9.14, and the best scores are presented in Figure 9.12. The results appear to mirror those found in Experiment 9, suggesting once again that looking 10 to 15 minutes before the fault occurs gives some of the better results, even when the data duration is only 10 minutes. Also similar to Experiment 9 there are some drops in AUC-ROC as the time until the fault occurs decreases. The cause of this can be explained in the same way as it was in Experiment 9.

Time until fault	Data duration	Classifier	F vs NF	GF vs NF	VS vs NF	PI vs NF
0 min	10 min	CatBoost	0.8433 (0.7804)	0.9559 (0.9300)	0.7231 (0.6330)	**1.0000 (0.9487)**
		FFNN	**0.8464 (0.7742)**	0.9619 (0.9200)	0.6614 (0.5931)	1.0000 (1.0000)
		RF	0.8131 (0.7595) MD=5,E=500	**0.9623 (0.9325) MD=15,E=250**	**0.7319 (0.6569) MD=25,E=400**	1.0000 (0.9744) MD=15,E=50
		SVM	0.8205 (0.7755) K=linear	0.9393 (0.9250) K=linear	0.6575 (0.5904) K=sigmoid	1.0000 (0.9744) K=linear
1 min	10 min	CatBoost	0.6793 (0.6245)	0.6797 (0.6400)	0.6322 (0.5878)	**0.8263 (0.7179)**
		FFNN	0.6121 (0.5877)	0.5593 (0.5600)	0.5857 (0.5691)	0.6263 (0.5641)
		RF	**0.7046 (0.6466) MD=None,E=200**	**0.6947 (0.6450) MD=15,E=50**	**0.6500 (0.5931) MD=None,E=150**	0.7079 (0.5897) MD=15,E=100
		SVM	0.4579 (0.5067) K=poly,D=2	0.4206 (0.5125) K=rbf	0.5000 (0.5080) K=sigmoid	0.4737 (0.7949) K=linear
5 min	10 min	CatBoost	0.6093 (0.5730)	0.6404 (0.6000)	0.5689 (0.5585)	**0.8645 (0.8205)**
		FFNN	0.5035 (0.5117)	0.5280 (0.5150)	0.4591 (0.4814)	0.4842 (0.4615)
		RF	**0.6441 (0.6049) MD=20,E=100**	**0.6907 (0.6400) MD=None,E=50**	**0.5907 (0.5585) MD=None,E=50**	0.8053 (0.7179) MD=15,E=50
		SVM	0.6121 (0.5767) K=linear	0.5505 (0.4550) K=sigmoid	0.5000 (0.5000) K=sigmoid	0.3737 (0.7436) K=linear
10 min	10 min	CatBoost	0.6462 (0.6012)	**0.7212 (0.6500)**	0.6748 (0.6170)	0.6789 (0.6154)
		FFNN	0.5799 (0.5620)	0.6210 (0.5225)	0.5672 (0.5372)	0.5684 (0.5128)
		RF	**0.6753 (0.6258) MD=None,E=500**	0.6873 (0.6200) MD=5,E=50	**0.6864 (0.6356) MD=None,E=350**	**0.7132 (0.6154) MD=5,E=100**
		SVM	0.5919 (0.5656) K=poly,D=2	0.5000 (0.5000) K=sigmoid	0.4754 (0.5000) K=sigmoid	0.6842 (0.6667) K=linear

Table 9.13: AUC-ROC scores for comparing balanced data-sets for various fault types using combined aggregated values on the 1kHz wave form data-sets presented in Tables 6.20, 6.21, 6.23, 6.24, 6.25, and 6.26. The best AUC-ROC score for each data-set is presented in bold, and the best score for each comparison across Tables 9.13 and 9.14 is highlighted in red, the accuracy is given in parentheses. Power interruptions versus non-faults is included for completeness, but due to the lack of instances of power interruptions, these scores ought to be considered an indicator more than a result.
The classifier parameters are abbreviated as: Max depth: MD, Estimators: E, Kernel: K, and Degree: D.
The fault types are abbreviated as: Faults: F, Non-faults: NF, Ground faults: GF, Voltage sags: VS, and Power interruptions: PI. 1/2

Time until fault	Data duration	Classifier	F vs NF	GF vs NF	VS vs NF	PI vs NF
15 min	10 min	CatBoost	0.6306 (0.5914)	**0.6781** (**0.6425**)	0.6153 (0.5824)	0.6961 (0.7179)
		FFNN	0.5549 (0.5374)	0.5467 (0.5125)	0.5250 (0.5213)	0.5974 (0.5385)
		RF	**0.6430** (**0.5926**) **MD=30,E=50**	0.6716 (0.6275) MD=10,E=150	**0.6814** (**0.6250**) **MD=25,E=250**	**0.8211** (**0.7179**) **MD=5,E=100**
		SVM	0.4880 (0.4945) K=poly,D=2	0.5332 (0.4825) K=sigmoid	0.4717 (0.4973) K=poly,D=3	0.7211 (0.7436) K=linear
30 min	10 min	CatBoost	0.6238 (0.6025)	0.6500 (0.5950)	0.5706 (0.5452)	0.6224 (0.6154)
		FFNN	0.5871 (0.5362)	0.6334 (0.5675)	0.4562 (0.4973)	**0.6316** (**0.5128**)
		RF	**0.6500** (**0.6049**) **MD=None,E=150**	**0.6715** (**0.6250**) **MD=None,E=150**	**0.6372** (**0.5984**) **MD=20,E=150**	0.6211 (0.5897) MD=5,E=450
		SVM	0.4727 (0.4994) K=poly,D=2	0.5025 (0.4975) K=linear	0.5000 (0.5399) K=poly,D=3	0.4974 (0.6410) K=linear
50 min	10 min	CatBoost	0.6363 (0.5963)	0.7276 (0.6650)	0.6358 (0.5851)	0.7395 (0.7692)
		FFNN	0.5520 (0.5276)	0.5004 (0.5125)	0.4552 (0.4867)	0.6237 (0.5897)
		RF	**0.6605** (**0.6294**) **MD=20,E=150**	**0.7442** (**0.6775**) **MD=30,E=50**	**0.6520** (**0.6064**) **MD=20,E=50**	0.7500 (0.6923) MD=25,E=100
		SVM	0.4810 (0.4994) K=poly,D=2	0.6894 (0.6275) K=rbf	0.4371 (0.5399) K=poly,D=2	**0.8684** (**0.8205**) **K=linear**

Table 9.14: AUC-ROC scores for comparing balanced data-sets for various fault types using combined aggregated values on the 1kHz wave form data-sets presented in Tables 6.20, 6.21, 6.23, 6.24, 6.25, and 6.26. The best AUC-ROC score for each data-set is presented in bold, and the best score for each comparison across Tables 9.13 and 9.14 is highlighted in red, the accuracy is given in parentheses. Power interruptions versus non-faults is included for completeness, but due to the lack of instances of power interruptions, these scores ought to be considered an indicator more than a result.
The classifier parameters are abbreviated as: Max depth: MD, Estimators: E, Kernel: K, and Degree: D.
The fault types are abbreviated as: Faults: F, Non-faults: NF, Ground faults: GF, Voltage sags: VS, and Power interruptions: PI. 2/2

9.11 Experiment 11

In Experiment 11 we examined how wavelet transform spectograms compared to wavelet scattering. The results are presented in Table 9.11. It is apparent that the CNN is not able to learn the data at all and achieves baseline performance. The reason may be that the spectograms are not able to capture any traits of the faults like the scattering is – as suggested in Section 7.6.2 – resulting in the CNN just classifying all the observations as one of the fault types. As scattering is both cheaper and faster to calculate, and is able to extract useful features, it seems like it is the superior feature extraction method of the two.

Freq	Data type	Features	Classifier	F vs NF	GF vs NF	VS vs NF	PI vs NF
1kHz	Wave form	Wavelet scattering	CatBoost	**0.5855** (**0.5570**) **J=15,Q=12**	0.6652 (0.6275) J=15,Q=24	**0.5948** (**0.6025**) **J=12,Q=96**	0.7675 (0.7250) J=6,Q=48
			FFNN	0.5230 (0.4917) J=6,Q=12	**0.6798** (**0.6275**) **J=8,Q=12**	0.5505 (0.5575) J=8,Q=12	**0.7734 (0.7000) J=15,Q=12**
			k-NN	0.5508 (0.5357) J=12,Q=12	0.6101 (0.5676) J=8,Q=12	0.5604 (0.5691) J=12,Q=12	0.7057 (0.6667) J=12,Q=24
1kHz	Wave form	WTS	CNN	0.500 (0.500)	0.500 (0.500)	0.500 (0.500)	0.500 (0.500)

Table 9.15: AUC-ROC scores for comparing balanced data-sets for various fault types using wavelet scattering and wavelet transform spectograms (WTS) on the 1kHz wave form data-set presented in Table 6.1. The best AUC-ROC score for each data-set is presented in bold, the accuracy is given in parentheses. Power interruptions versus non-faults is included for completeness, but due to the lack of instances of power interruptions, these scores ought to be considered an indicator more than a result.
The fault types are abbreviated as: Faults: F, Non-faults: NF, Ground faults: GF, Voltage sags: VS, and Power interruptions: PI.
Wavelet transform spectograms are abbreviated as WTS.

Chapter 10

Future Work

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

10.1 Improving the Labeling Scheme

In Section 7.1.1 the current labeling scheme used by DDG was discussed, and some improvements were suggested. Implementing an improved labeling scheme can give a better basis for classification, especially when comparing different types of faults.

10.1.1 Fault Overlap and Fault Sequences

Once an improved labeling scheme has been implemented, fault overlap times and fault sequences discussed in Sections 7.1.1 and 7.1.2 should be revisited, especially the tendency of ground faults and voltage sags to overlap and lead into each other.

10.2 Time and Date Features

In Section 7.1.3 we noted that some fault types appear to happen more frequently at some times of day and year. There might be other tendencies for when different faults occur, and exploring this further might give access to information which can be used to improve

accuracy of models. There might be knowledge to gain in redoing this exploration for individual nodes, as some might be more susceptible to seasonal changes than others.

10.3 Node Specific Learning

As discussed in Section 7.1.4, the distribution of faults is skewed between the different nodes. As further explored in Section 7.5, the difference between nodes also greater than that between faults. Due to this we suggest that learning to classify faults for specific nodes might yield better results than trying to learn a general model for all of them, as a general model will become subject to bias.

10.3.1 Synthetic Data Generation

As few nodes report enough faults to achieve a well trained model on them alone, synthesising data might be necessary, especially if a balanced data-set is desired.

10.3.2 Transfer Learning

If synthesising data is not desirable, creating a transfer learning environment might be a good alternative. In the transfer learning environment the model would first learn on all nodes – as we have done in this thesis – but after having been trained, it would be further trained on data solely from the node in focus. By doing this the model will learn general characteristics of fault types first – with the node bias discussed – to then later try to remove the associations between nodes and fault types, leaving a model which only classifies on characteristics of fault types.

10.3.3 Further Exploration of Node Characteristics

Doing further exploration of what characterises the different nodes might reveal the impact of, and new ways to combat, the bias discussed in Section 7.5, where fault types such as power interruptions might be linked to characteristics of Node2.

10.4 Wavelet Scattering

10.4.1 Bigger Parameter Scope

For the 1kHz data-sets, which had 60,000 samples per observation, we were able to try 4x4=16 different combinations of J and Q. However for the 25kHz data-sets, with 60,000x25=1,500,000 samples per observation, we did not have time as making wavelet scattering coefficients for each combination of J and Q took a significantly long time to calculate. We limited the scope by choosing the combination that had the best performance averaged over all the classifiers for 1kHz. To fully explore the potential of wavelet scattering on the 25kHz data-sets we therefore suggest an extensive parameter search trying a bigger variety of Js and Qs.

10.4.2 Optimizing for Real-time

As of now the wavelet scattering had to be calculated for each interval in order to predict whether or not a fault is going to occur. This is very costly and might not be possible to do constantly, and pauses might have to be inserted between the calculations, especially for high frequency signals with duration over one minute. To make this more effective one could for instance try to aggregate the coefficients over smaller time intervals in the same fashion that we aggregated the wave signal. This would make the calculation a lot faster as for every time we want to predict a fault, we only have to calculate the scattering for the new unseen time interval, and we can reuse the previous calculated coefficients instead of recalculating everything in order to cover the new data.

One could also try to sample smaller parts of the wave signal over a larger time interval in order to get a wider coverage of samples without increasing the amount. This might be a drawback as each part would be considerably shorter that just looking at one continuous interval, and the fault might only be detectable looking at longer continuous intervals. This would also create a lot of noise in the intersections where the segmented signal transitions from one interval to another as it would not be continuous.

10.5 Data

In this thesis we only used phase-to-ground phases and not the phase-to-phase phases. These should also be considered included for a higher chance of finding traits unique to the faults.

10.6 Other Aggregation Methods

In this thesis we used very simple aggregation methods as features such as the max, min and mean value. Other aggregation methods as mentioned in Section 5.3 should be considered.

10.7 Other Models

While singular aggregated values score consistently worse than combined aggregated values, they did show a tendency which suggests there might be more information to be found when looking at smaller windows of time prior to the fault. In this thesis we used models which were not suited to consider temporal changes, which is why the original data-set containing 45 values over 60 seconds was changed as described in Section 6.3.3. Using a model which can take temporal changes into account, so that the original data can be used without modification, might remove the tendency of overfitting described in Experiment 2, while additionally retaining the ability to let noise affect the calculated aggregated values, as described in Experiment 6.

10.8 Weighted Sampling

In this thesis we attempted to sample each second for one minute one minute before the fault, and one second each minute an hour leading up to the fault. Other sampling methods should be tested out, one suggestion is a weighted sampling method, where the duration between each sample gets longer as the samples gets farther away from when the fault occurs. One example could be to sample every second one minute prior to the fault, then every other second the minute before that, every third second the minute before that, and so forth.

Chapter 11

Conclusion

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

RQ1: To what extent do there exist differentiable structures in the data?

Section 7.5 revealed the most differentiable structure in the data to be the origin node. When looking at the wave up until the fault for a single node, ground faults and power interruptions were both found to be noticeably unique, and separated from the rest of the observations. Voltage sags did not share this characteristic to as high a degree, and tended to group up with non-faults. This relation came up again in our experiments, where voltage sags tended to be the hardest to differentiate from non-faults, even in Experiments 9 and 10, which included data up until the occurrence of the fault.

The fact that the most differentiable structure in the data is the origin nodes, suggests that the classifying models used in the EarlyWarn project ought to take into account the origin of the retrieved data, ways to do this was suggested in Section 10.3. The faults reported and the distribution of the reported faults for each node vary significantly, as noted in Section 7.1.4. This when combined with the apparent differences between the different nodes, can create significant bias in classifiers if not taken into consideration.

RQ2: Which data representations are the most useful for predicting faults in the power grid?

There are many different ways of representing wave signals, the Fourier transform is one natural choice as the data in theory should be a stationary sinus wave with noise in the form of harmonics which can be used to identify the occurring faults. The wavelet transform is another representation which in some ways is an improvement of the Fourier transform and has shown a lot of promise as discussed in Chapter 5. There is also the option of using the raw wave as is, and the RMS values. In regard to the raw wave a sampling frequency

has to be chosen which will decide how accurately the wave is represented.

Our results from Experiments 1 through 3 showed that aggregating values based on the raw waveform, rather than the Fourier transform and RMS values of the wave, gave the best results. Experiment 4 revealed that aggregation, rather than wavelets, gave better results. Experiment 6 showed some positive impact of sampling at higher frequencies, but for the noise which is present in higher frequencies to be captured in the aggregated values, more suited aggregation methods than max, min, mean, STD and SNR is needed. In Experiment 3 it was shown that looking at fewer samples of the wave over one hour contains better classifying information than looking at the full minute one minute before the occurrence of the fault.

In sum, the data representation which is most useful for predicting faults in the power grid appears to be aggregated values of the wave form over longer time intervals at high frequencies, given that the aggregated values manage to express the presence of eventual noise, or are invariant to frequency when this is not the case.

RQ3: How long before faults occur does the signal contain information which differentiates them from normal behavior?

To be able to predict the faults it is useful to know when the faults first start to appear in the wave signal and if there are some specific time intervals in which they are more visible. The results from Experiments 9 and 10 showed that it is somewhat possible to predict faults looking at the signal up to 50 minutes before the fault occurred, but that it got easier to predict as data from time intervals closer to the fault were included. Based on this, it seems like some faults start to show themselves at least 50 minutes before they occur, but that the strongest traits appear closer to the actual occurrence of the faults. This means that even though it is not reliable to predict faults looking only at time intervals long before the fault occurs, those intervals still contain useful information. This could be better taken advantage of by using a weighed scheme as suggested in Section 10.8.

RQ4: What prediction performances are achievable using machine learning methods?

We evaluate the most promising machine learning methods discussed in Chapter 5: Random forest, *k*-NN, CatBoost, SVM, FFNN and CNN. Looking at the results from all the experiments done, random forest and CatBoost have had the overall best performances. SVM has without question performed the worst. Feedforward neural network have had some okay scores, but have mostly been outshone by random forest and CatBoost. For Experiments 1 through 4 all of the classifiers had one top score each, even though SVM's top score was for power interruptions which is not that trustworthy because of the lacking data-set size. Looking at different frequencies in Experiments 5 through 8, and different forecast horizons in Experiments 9 and 10, the top scores were mostly split between random forest and CatBoost. For wavelet scattering, CatBoost and feedforward neural network performed the best. For the wavelet transform spectograms we only tried CNN, but as the wavelet transform was not able to capture any useful features we cannot say much about its performance.

Random forest achieved the best AUC-ROC of 0.7063 and an accuracy of 0.6488 for faults versus non-faults.

Bibliography

Albelwi, S., Mahmood, A., 2017. A Framework for Designing the Architectures of Deep Convolutional Neural Networks. Entropy .

Andresen, C.A., Torsaeter, B.N., Haugdal, H., Uhlen, K., 2018. Fault Detection and Prediction in Smart Grids. 9th IEEE International Workshop on Applied Measurements for Power Systems, AMPS 2018 - Proceedings .

Andreux, M., Angles, T., Exarchakis, G., Leonarduzzi, R., Rochette, G., Thiry, L., Zarka, J., Mallat, S., Andén, J., Belilovsky, E., Bruna, J., Lostanlen, V., Hirn, M.J., Oyallon, E., Zhang, S., Cella, C., Eickenberg, M., 2018. Kymatio: Scattering Transforms in Python. arXiv:1812.11214.

Ataspinar, A., 2018. A guide for using the Wavelet Transform in Machine Learning. http://ataspinar.com/2018/12/21/a-guide-for-using-the-wavelet-transform-in-machine-learning/. Accessed: 2020-05-24.

Chepenko, D., 2019. Introduction to gradient boosting on decision trees with CatBoost. https://towardsdatascience.com/introduction-to-gradient-boosting-on-decision-trees-with-catboost-d511a9ccbd14/. Accessed: 2020-06-16.

Chollet, F., et al., 2020. Keras. https://keras.io.

Devleker, K., 2016. Understanding wavelets (Mathworks). https://www.mathworks.com/videos/series/understanding-wavelets-121287.html. Accessed: 2020-04-19.

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.

Gaouda, A., Salama, M., 2009. Monitoring Nonstationary Signals. Power Delivery, IEEE Transactions on 24, 1367 – 1376. doi:10.1109/TPWRD.2009.2013386.

Goodfellow, I., Bengio, Y., Courville, A., 2016. Deep Learning. MIT Press.

Gopakumar, P., Reddy, M.J.B., Mohanta, D.K., 2015. Transmission line fault detection and localisation methodology using PMU measurements. IET Generation, Transmission Distribution 9, 1033–1042.

Hoffmann, V., Michałowska, K., Andresen, C., Torsaeter, B., 2019. Incipient Fault Prediction in Power Quality Monitoring. International Conference on Electricity Distribution .

Hunter, J.D., 2007. Matplotlib: A 2D graphics environment. Computing in Science & Engineering 9, 90–95. doi:10.1109/MCSE.2007.55.

Høiem, K.W., 2019. Predicting Fault Events in the Norwegian Electrical Power System using Deep Learning. Master's thesis. Norwegian University of Life Sciences (NMBU).

Jahr, C., Meen, H.K., 2019. Predicting faults in power grids using machine learning methods. https://github.com/Cami-Jahr/Specialisation-Project/blob/master/NTNU_Specialization_Project.pdf. Accessed: 2020-04-09.

Kaggle, 2019. VSB Power Line Fault Detection. https://www.kaggle.com/c/vsb-power-line-fault-detection/overview. Accessed: 2020-05-24.

Kehtarnavaz, N., 2008. Digital Signal Processing System Design. Academic Press.

Lee, G.R., Gommers, R., Wasilewski, F., Wohlfahrt, K., O'Leary, A., 2019. PyWavelets: A Python package for wavelet analysis. Journal of Open Source Software URL: https://doi.org/10.21105/joss.01237.

Li, L., 2019. Classification and Regression Analysis with Decision Trees. https://medium.com/lorrli/classification-and-regression-analysis-with-decision-trees-c43cdbc58054/. Accessed: 2020-06-16.

Lundh, F., Clark, A., 1995. Pillow. https://pillow.readthedocs.io/en/stable/index.html. Accessed: 2020-10-06.

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

Mahela, O.P., Shaik, A.G., Gupta, N., 2015. A critical review of detection and classification of power quality events. Renewable and Sustainable Energy Reviews 41, 495–505. doi:10.1016/j.rser.2014.08.070.

Mallat, S., 2012. Scattering Invariant Deep Networks for Classification. http://helper.ipam.ucla.edu/publications/gss2012/gss2012_10668.pdf. Accessed: 2020-04-19.

mark4h, 2019. Overview of 1st place solution. https://www.kaggle.com/c/vsb-power-line-fault-detection/discussion/87038. Accessed: 2020-05-24.

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

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

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library, in: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alch'e Buc, F., Fox, E., Garnett, R. (Eds.), Advances in Neural Information Processing Systems 32. Curran Associates, Inc., pp. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

Patel, A., 2012. FeedForward Neural Network and Back Propagation. https://mc.ai/chapter-2-3-deep-learning-101-feedforward-neural-network-and-back-propagation/. Accessed: 2020-04-19.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E., 2011. Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research 12, 2825–2830.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A.V., Gulin, A., 2017. CatBoost: unbiased boosting with categorical features. arXiv:1706.09516.

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

Santi, V.M., 2019. Predicting faults in power grids using machine learning methods. Master's thesis. Norwegian University of Science and Technology (NTNU).

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.

Sharma, S., 2019. Kernel Trick in SVM. https://medium.com/analytics-vidhya/how-to-classify-non-linear-data-to-linear-data-bb2df1a6b781/. Accessed: 2020-06-16.

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

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

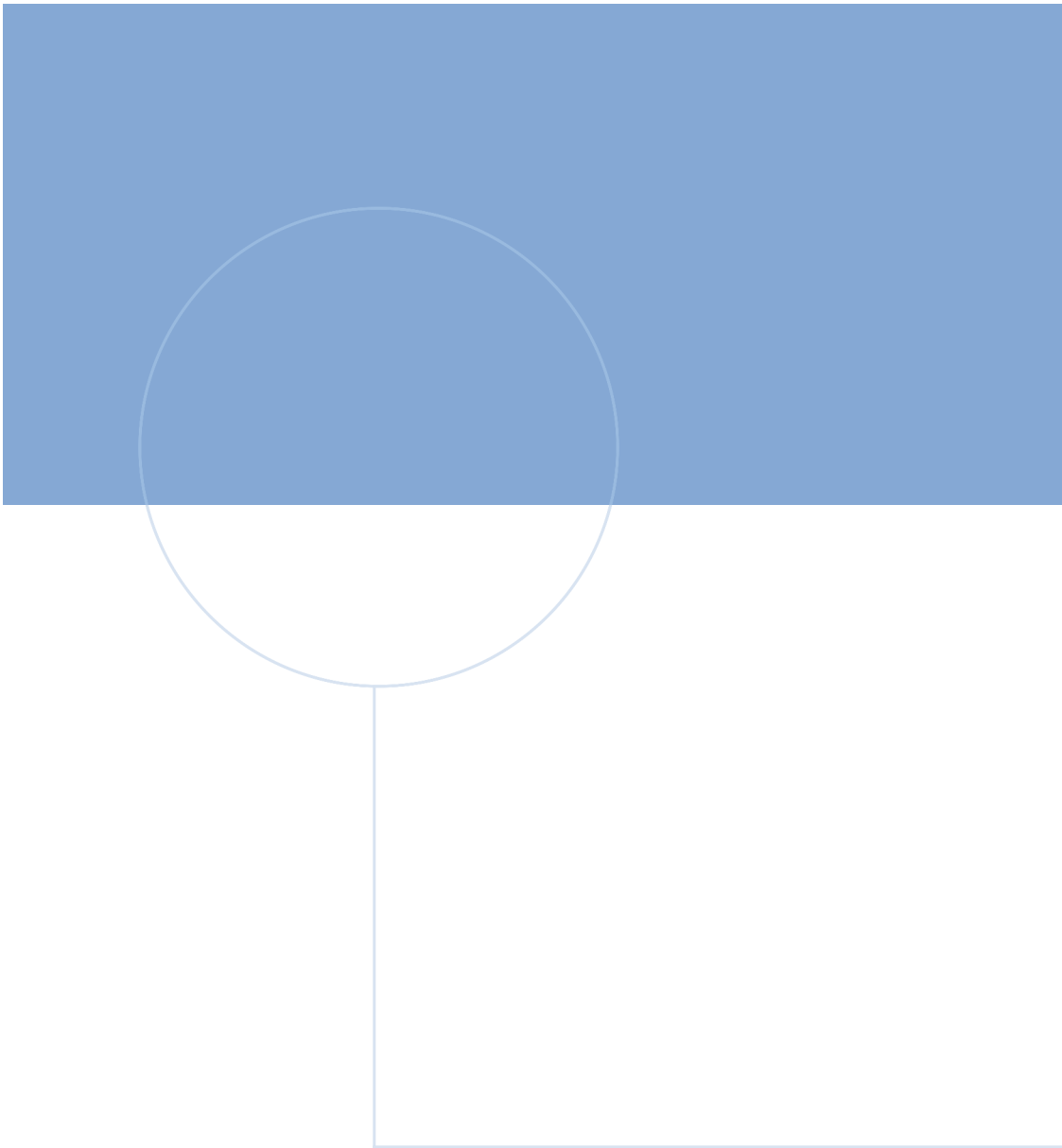
Statistisk Sentralbyrå, 2019b. Kraftforsyning bidro til investeringsvekst i 2018. https://www.ssb.no/energi-og-industri/artikler-og-publikasjoner/kraftforsyning-bidro-til-investeringsvekst-i-2018. Accessed: 2019-12-03.

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 .

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



NTNU
Norwegian University of
Science and Technology

SINTEF