Vemund Mehl Santi

Predicting faults in power grids using machine learning methods

Master's thesis in Computer Science
Supervisor: Helge Langseth, Bendik Torsæter, Volker Hoffmann
June 2019

Master's thesis

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

NTNU
Norwegian University of
Science and Technology

SINTEF

Vemund Mehl Santi

Predicting faults in power grids using machine learning methods

Master's thesis in Computer Science
Supervisor: Helge Langseth, Bendik Torsæter, Volker Hoffmann
June 2019

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

In a world that is increasingly dependent on electricity, providing a stable power distribution network is of utmost importance. With the recent advances in smart grid technology, actors that used to be pure consumers are now becoming both producers and consumers of energy. This change leads to a more rapid change in load on the electrical grid, with variable load at uneven intervals. The changes call for better monitoring and analysis tools, to cope with the added dynamicity from thousands of new producers.

In this thesis we examine the field of power system fault prediction, the task of predicting a fault ahead of time. Specifically we implement multiple machine learning models that take as input voltage measurement data, collected from Power Quality Analysers in the Norwegian power grid.

We present multiple feature engineering methods that aggregates time series of harmonic frequencies in high resolution voltage data. The features are combined with Support Vector Machines, Random Forests and multiple Neural Network architectures, with the aim of predicting faults with a 10 minute horizon. Our models are able to successfully predict over 74 % of faults within the fault categories voltage sags, interruption faults, ground faults and rapid voltage change faults.

Keywords Norwegian Power grid, Power Quality Analyzers, Machine learning, Voltage analysis, Fault prediction

Sammendrag

I en verden som er stadig mer avhengig av elektrisitet, er det av svært viktig å sørge for et stabilt kraftdistributionsnett. Med de siste fremskrittene innen smart grid teknologi, er aktører som tidligere pleide å være rene forbrukere nå blitt både produsenter og forbrukere av energi. Denne endringen fører til en raskere endring i belastningen på kraftnettet, med store endringer i belastningsgraden med ujevne mellomrom. Endringene krever bedre overvåkning- og analyseverktøy for å takle den økte dynamikken fra tusenvis av nye produsenter.

I denne masteroppgaven utforsker vi feltet for feilhendelsesforutsigelse i strømnettet, det å forutsi en feil før den inntreffer. Vi implementerer flere maskinlæringsmodeller som benytter spenningsmålinger fra strømkvalitetssensorer i det norske strømnettet.

Vi presenterer flere funksjonsekstraksjonsmetoder som aggregerer tidsserier av harmoniske frekvenser i høyoppløselig spenningsdata. Data brukes i kombinasjon med Support Vector Machines, Random Forests og flere ulike Neural Network arkitekturer, med mål å forutsi feil med 10 minutters tidshorisont. Våre modeller er i stand til å forutsi over 74% av feilene i feilkategoriene spenningsdipp, avbrudd, jordfeil og sprang.

Acknowledgements

I would like to thank my supervisor Helge Langseth at NTNU for invaluable support and feedback during the writing of this thesis. Your feedback has been a great source of input and motivation, and I thank you for taking the time to provide weekly feedback during the semester.

I would also like to extend my thanks to my co-supervisors, fellow project members and project leader at SINTEF Energy, Bendik Nybakk Torsæter, Gjert Hovland Rosenlund, Volker Hoffmann and Christian Andresen. Your expertise and endless source of information in the power domain has been of great help.

Table of Contents

List of Tables

List of Figures

Abbreviations

Symbol	=	Definition
AC	=	Alternating Current
AI	=	Artificial Intelligence
AUC	=	Area Under the Curve
CENS	=	Cost of Energy Not Supplied
CLI	=	Command Line Interface
CV	=	Cross-validation
DC	=	Direct Current
DDG	=	Dynamic Dataset Generator
DSO	=	Distribution System Operator
DT	=	Decision Trees
FFT	=	Fast Fourier Transform
HIF	=	High Impedance Faults
HMM	=	Hidden Markov Models
I	=	Current
LSTM	=	Long Short-Term Memory
MCC	=	Matthews Correlation Coefficient
NN	=	Neural Networks
NTNU	=	Norwegian University of Science and Technology
PMU	=	Phasor Measurement Units
PQ	=	Power Quality
PQA	=	Power Quality Analyzer
PQI	=	Power Quality Indices
ReLU	=	Rectified Linear Unit
RMS	=	Root Mean Square
RNN	=	Recurrent Neural Network
ROC	=	Receiver Operating Characteristic
RVC	=	Rapid Voltage Change
STFT	=	Short Time Fourier Transform
SVM	=	Support Vector Machines
THD	=	Total Harmonic Distortion
TSO	=	Transition System Operator
V	=	Voltage
WT	=	Wavelet Transform

Chapter 1

Introduction

This Master's Thesis is part of the ongoing research project EarlyWarn, a collaboration project between SINTEF Energi, SINTEF Digital, NTNU, Statnett, Haugaland Kraft Nett, NTE Nett, Lyse Elnett, Nettalliansen and Hydro Energi. The project is part of the ENERGIX programme and partly financed by the Research Council of Norway. Their ultimate goal is proactive detection and early warning of incipient power grid faults and instabilities by using data from sensors placed at strategic places in the Norwegian power grid [1].

The Master's project has consisted of two parts: a pre-study conducted during the autumn of 2018 in collaboration with Vegard Hellem, and the final Master's Thesis due June 2019. The final thesis build on the pre-study conducted in 2018, which serves as the basis for Chapter 1, 2, 3, 4 and 5.

In Section 1.1 we will introduce the motivation behind this master's thesis, and the recent advances in technology enabling this project. We will follow up on this motivation and the problems presented in Section 1.2 by formulating a set up research questions that are answered in this thesis. Finally, Section 1.3 gives an outline of how this thesis is structured.

1.1 Motivation

The power grid is one of the most important parts of infrastructure in the modern world and it is hard to imagine a world without electricity. The total use of electricity per capita in Norway is one of the highest in the world, primarily because electric power is commonly used to heat residential and industrial buildings [2]. Researchers have recently looked at how to make the grid more robust and secure and are moving towards a Smart Grid

architecture to enable more flexible solutions [3]. In Norway, the grid includes over 130 000 km of power lines, divided between the transmission, regional and distribution grid [4]. The net power consumption in Norway in 2016 was 116.6 TWh [5], and the energy consumption is expected to increase in the future [6].

The first part of the Norwegian power grid was built in 1892 and powered a single light bulb in Oslo [7]. Construction thereafter expanded, and large parts of the current infrastructure were built in the 1960s and 1970s. These parts are now reaching the end of their expected lifetime, and the grid in its current state requires substantial renovation, upgrades, and replacements [8].

Since the middle of the 2000s, large investments have been made to expand and improve the power grid infrastructure. In 2017, the total investment in the power supply infrastructure was 32.7 billion NOK, and it is estimated a 25.8% increase to 41.1 billion NOK for 2018 [9]. The recent years' investment cost can be seen in Figure 1.1.

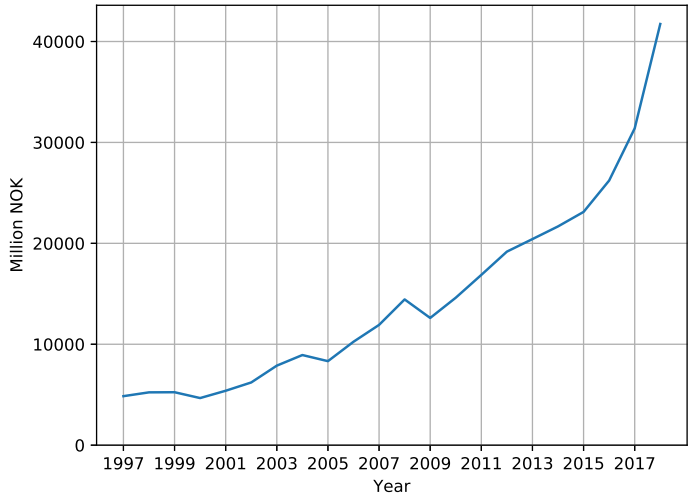


Figure 1.1: Total cost per year for investments in the Norwegian power grid 1997-2018 [8, 9].

With the combination of aging infrastructure, Norway's hazardous weather, and the sheer size of the power grid, interruptions naturally occur and come with a cost to repair. In 2017, 63% of end-users in Norway were affected by faults, experiencing one or more interruptions lasting longer than 3 minutes [5]. An average of 793 faults occurred every year between 2008 and 2017 in the Norwegian power grid [10], causing a mean 5 042 MWh of power not to be delivered to consumers. This amounts to only 0.12‰ of the total delivered power, but a cost of 800 million NOK per year in maintenance and repairs [11]. This cost is known as the CENS cost (Cost of Energy Not Supplied). Figure 1.2 shows an overview of CENS costs per county in Norway. If one could anticipate some

of these faults, it could have a substantial socioeconomic impact in reducing the costs of maintaining the power grid.

Recent advances in sensors and monitoring of the power grid make such predictions possible. In Norway, there are Power Quality Analyzers (PQA) and Phasor Measurement Units (PMU) in place at strategic sites around the grid, that monitor and send data to a centralized server. This data can be combined with domain knowledge of disturbances in the power grid and machine learning techniques. The result could make it possible to give an early warning of upcoming faults.

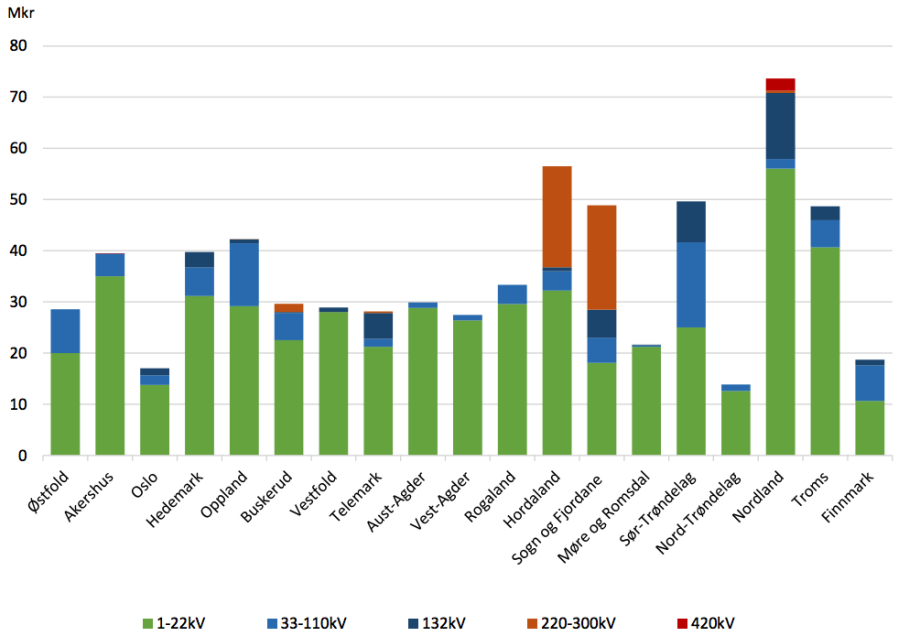


Figure 1.2: An overview of CENS costs for counties in Norway. All amounts in million NOK.

1.2 Research questions

We have formulated the following research questions:

RQ 1. Can machine learning methods predict upcoming faults on power lines by analyzing voltage measurement data?

RQ 2. What attributes in a voltage signal are suited for predicting faults on power lines?

RQ 3. What is the performance of some machine learning algorithms at predicting faults on power lines?

RQ 4. Are some types of faults easier to predict than others?

1.3 Thesis structure

The thesis is structured as follows: Chapter 2 and Chapter 3 introduces the background theory required for understanding the domains of power analysis and machine learning used in this thesis. Chapter 4 contains a summary of previous work in the field of power fault analysis, while Chapter 5 introduces the EarlyWarn project, which this thesis is a part of. Chapter 6 describes in detail the experiments that were carried out in order to answer the research questions from Section 1.2 and lists the results of each experiment. Chapter 7 analyses the finding from the experiments, discusses their significance and recommend further research areas based on the results. Finally, Chapter 8 concludes the project conducted in this report.

Chapter 2

Background - Power Grids

The contents of this chapter is based on a semester project conducted in collaboration with Vegard Hellem during autumn 2018. The project report is available online [12].

This chapter describes common terms and methods used for describing power and power quality. Section 2.1 describes common current and voltage metrics, methods used for power analysis and three-phase electric power. Section 2.2 looks at the most common faults and disturbances in power grids around the world and the reason for faults in the Norwegian power grid.

2.1 Power

2.1.1 Current and Voltage

Definitions

Current (I) is the rate of flow of electric charge past a given point [13]. Current is measured in ampere (A), where 1 ampere is defined as a flow of 1 coulomb of charge past the given point.

Voltage is the cost in energy required to move a unit of positive charge from a point with a lower electric potential to a point with higher electric potential [13]. The unit of measure is volt (V).

Direct and alternating currents (DC and AC)

Alternating current (AC) is the type of current most commonly used in electric grids throughout the world [14]. In contrast to direct current (DC), alternating current switches the direction of the current flow periodically, while direct current maintains a constant direction of current flow. In Norway, the standard AC frequency in the electric grid is 50Hz [15]. A plot comparing direct and alternating current is shown in Figure 2.1.

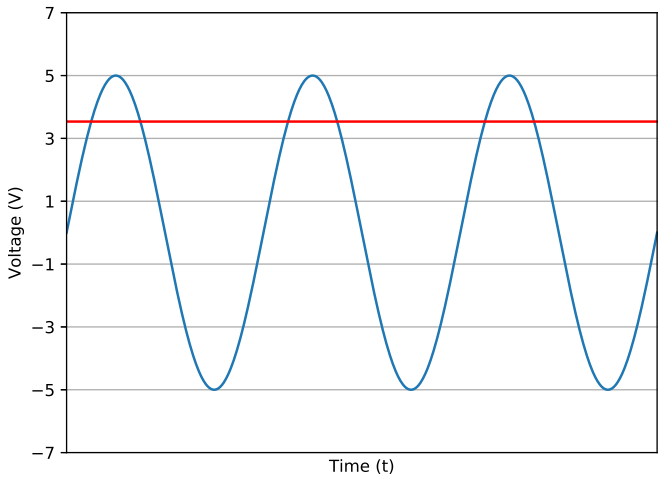


Figure 2.1: Example of a direct voltage level (red) and alternating voltage level (blue).

One of the advantages of using systems with AC over DC is the ability to change the voltage level through the use of transformers easily. Everyday household electronics in Norway needs a voltage level of 230V[15], but a much higher voltage level is beneficial when transmitting electricity across large distances. The ability to increase the voltage level before transmission and decrease the voltage level before distribution is a crucial advantage of using AC.

Alternating current as a mathematical function

A periodic, continuous alternating current can be described by a sinusoidal function of the following form [13]:

$$v(t) = \alpha \sin(\omega t + \phi) \tag{2.1}$$

where α is the maximum amplitude, ω is the angular frequency, t is a given point in time and ϕ is the initial phase of the signal. Figure 2.2 visualizes the relationship between the sinusoidal function and a periodic, alternating current.

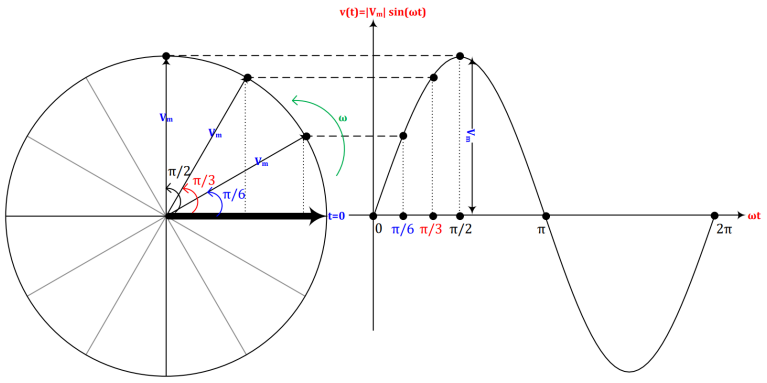


Figure 2.2: The relationship between a sinusoidal function (right) and an alternating current (left). V_m is the maximum amplitude of the alternating signal, ω is the angular frequency and t is time [16].

2.1.2 Power analysis

When conducting research and analysis of electric circuits, a wide variety of methods and measurements are used. In this section we will describe the following methods and measurements, which are commonly used in signal processing today:

1. Steady state periods
2. Root Mean Square (RMS)
3. Fourier analysis
4. Discrete Fourier analysis
5. Short Time Fourier Transform
6. Harmonics
7. Total Harmonic Distortion

Steady state periods

The steady state of a sinusoidal power function is a state where the sinusoidal signal repeats with a constant period T and does not change frequency nor amplitude between periods.

If the sinusoidal representation of the signal fulfills the following equality, we say that the signal is in a steady state [17]:

$$\alpha \sin(\omega t) = \alpha \sin(\omega t + nT) \tag{2.2}$$

where n is an arbitrary integer.

A power system should generally be in a steady state condition, but could after a disturbance, or when starting up, be in a transient condition. A healthy power line should after a short period return to steady state condition to prevent errors and damages to the equipment [17]. In practice, there is no such thing as a mathematically perfect steady state, as there are countless small disturbances affecting the power line in the real world. When we use the term steady state on real power lines, we refer to a state where the frequency and amplitude of the signal is close to its nominal values.

Root Mean Square (RMS)

The Root Mean Square is a standard measurement used for power quality in AC systems [18]. The value is a way of characterizing sinusoidal functions through a mean value, and thus reducing the complexity of comparison functions. The RMS voltage of an AC signal is equal to the voltage of a DC system that transfers the same amount of power [19]. The RMS value of a sinusoidal function $v(t)$ is defined as

$$V_{RMS} = \sqrt{\frac{1}{T}\int_0^T v(t)^2 dt} \tag{2.3}$$

where $v(t)$ is a sinusoidal function with period T. By expanding the sinusoidal function and using the trigonometric formula [19]

$$\cos^2(\omega t) = \frac{1}{2}(1 + \cos(2\omega t)) \tag{2.4}$$

we can simplify the equation to a simple expression:

$$
\begin{aligned}
V_{RMS} &= \sqrt{\frac{1}{T}\int_0^T v(t)^2 dt} \\
&= \sqrt{\frac{1}{T}\int_0^T \alpha^2 \cos^2(\omega t) dt} \\
&= \sqrt{\frac{\alpha^2}{T}\int_0^T \frac{1}{2}(1+\cos(2\omega t)) dt} \\
&= \sqrt{\frac{\alpha^2}{2T}T} \\
&= \frac{\alpha}{\sqrt{2}}
\end{aligned}
\tag{2.5}
$$

where α is the amplitude of the signal and $\int_0^T \cos(2\omega t)dt$ is zero, because T is the period of the sinusoidal function [19].

Figure 2.3 shows the relationship between the amplitude of an AC signal and its RMS value. Deviations in the RMS value of a signal is often used to detect faults in the power grid, for example by Sintef's AHA (Automatisk Hendelsesanalyse) software [20]. One often assumes that as long as the RMS voltage remains constant, the system is in a steady state. Note, however, that not all faults and disturbances in the grid can be detected by only monitoring the RMS value [21].

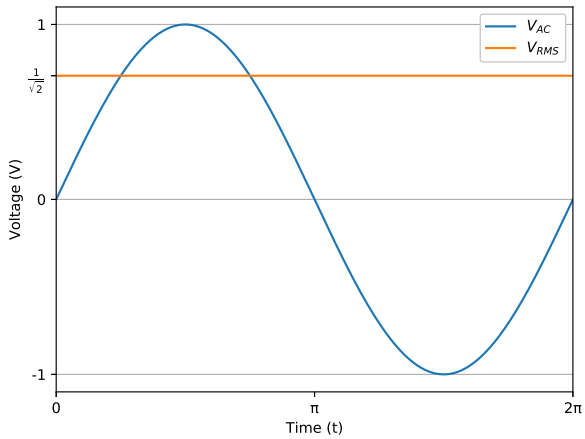


Figure 2.3: AC voltage level (blue) and corresponding RMS voltage level (orange) of a signal with amplitude 1 and period 2π.

Fourier analysis

Fourier analysis is the study of approximating functions with a finite sum of sinusoidal functions. It is often used in signal processing to decompose signals into separate frequencies, through the use of a Fourier Transform.

A Fourier Transformation is a function $\mathscr{F}$ that takes as input an integrable function, usually a time-based signal, and outputs a function which maps frequencies to Fourier coefficients. By summing all possible frequencies multiplied with their Fourier coefficients, we can restore the original function which was given as input to the Fourier Transformation $\mathscr{F}$ [22].

A continuous Fourier Transform is defined as [23]:

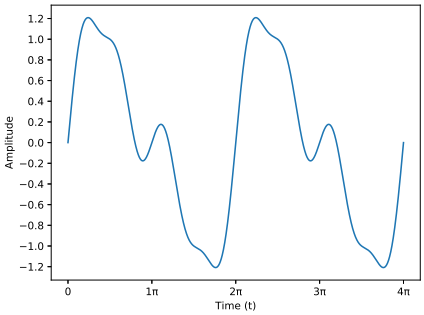
$$\hat{x}(f) = \mathscr{F}(x(t)) = \int_{-\infty}^{+\infty} x(t) e^{-i2\pi ft} dt \tag{2.6}$$

where $\hat{x}(f)$ is a function that outputs the Fourier coefficient for an input frequency f. The original signal function is given by $x(t)$ which takes as input a point in time t, and i is the imaginary unit. One can describe the Fourier Transform as transforming a function from the time domain to a function in the frequency domain.

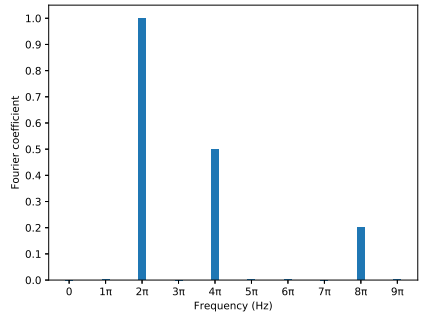
The Fourier Transform is invertible and can thus be used to map frequencies into a time-dependent signal again. The following equation defines the inverse continuous Fourier Transform $\mathscr{F}^{-1}$ [23]:

$$x(t) = \mathscr{F}^{-1}(\hat{x}(f)) = \int_{-\infty}^{+\infty} \hat{x}(f) e^{i2\pi ft} df \tag{2.7}$$

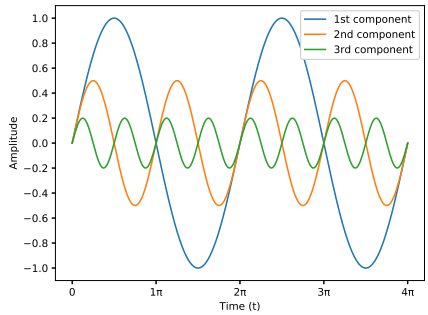
The relationship between a periodic, sinusoidal signal and it's Fourier Transform is shown in Figure 2.4 and 2.5



(a) Sinusoidal signal with a period of 2π.

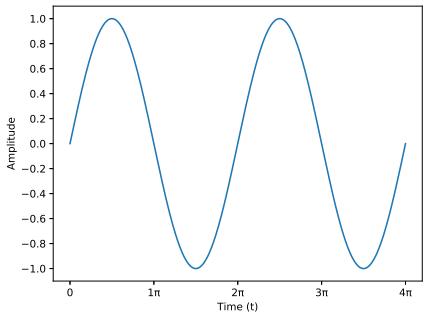


(b) The Fourier coefficients of the periodic signal in 2.4a.

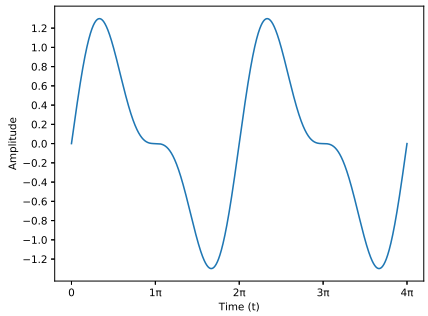


(c) The individual sine components that make up the signal in 2.4a.

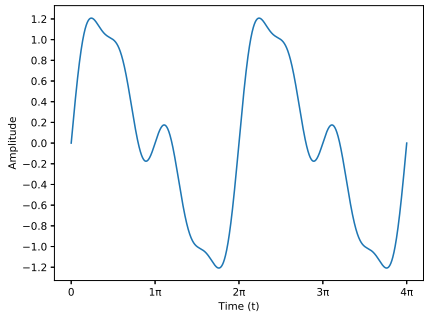
Figure 2.4: Three plots of a periodic signal, it's Fourier coefficients and its corresponding sine components. The periodic signal is given by $\sin(2\pi t) + 0.5\sin(4\pi t) + 0.2\sin(8\pi t)$. Note that the Fourier coefficients are 1, 0.5 and 0.2 at the frequencies 2π, 4π and 8π respectively.



(a) The first sine component of the signal in 2.4a.



(b) The sum of the first and second sine components of the signal in 2.4a.



(c) The sum of the first, second and third sine components of the signal in 2.4a.

Figure 2.5: Three plots showing the gradual composition of individual sine components from signal 2.4a. Notice that the sum of all the components is equal to the original signal.

Discrete Fourier analysis

When performing signal analysis in the real world, the signal $x(t)$ is not recorded as a continuous function, but rather as a series of samples with a given interval I. Each sample k can be regarded as an impulse with area $x[k]$, written as

$$x[k] = \int_{-\infty}^{+\infty} \delta(t-k)x(k)dt \tag{2.8}$$

where $\delta(t-k)$ is the Dirac Delta function [24] defined as

$$\int_{-\infty}^{+\infty} \delta(t-k)dt = \begin{cases} 1 & \text{if } t = k \\ 0 & \text{if } t \neq k \end{cases} \tag{2.9}$$

The area will then be zero between each sample, and we can transform the definition of the Fourier Transform from an integral into a sum

$$\begin{aligned} \mathscr{F}(x(t)) &= \int_{-\infty}^{+\infty} x(t)e^{-i2\pi ft}dt \\ &= x[0] \cdot e^{-i2\pi f0} + ... + x[k] \cdot e^{-i2\pi fk} + ... + x[N-1] \cdot e^{-i2\pi f(N-1)} \\ &= \sum_{k=0}^{N-1} x[k]e^{-i2\pi fk} \end{aligned} \tag{2.10}$$

where N is the number of samples [25].

Short Time Fourier Transform

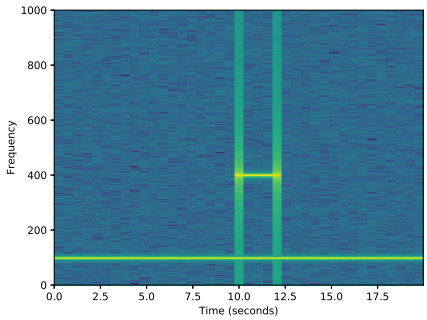
The drawback of using the Fourier Transform to analyze periodic signals is that it completely obscures the temporal locality of the signal. In some applications, it might be beneficial to retain some of the information from the time domain, with a trade-off of some loss in frequency resolution. The Short Time Fourier Transform introduces such a compromise.

By defining av window of length L along the time dimension of the signal, and computing the Fourier Transform of the signal within this window only, we transform the signal from the time domain into the frequency domain, while still keeping the transformation bound to an interval in the time domain. By moving the window along the time dimension of the original signal, we can transform each segment into its frequency domain, and the result would be a 3-dimensional tensor with time segments, frequencies, and amplitude as its dimensions. The transformation for the signal given by $x(t)$ is defined as

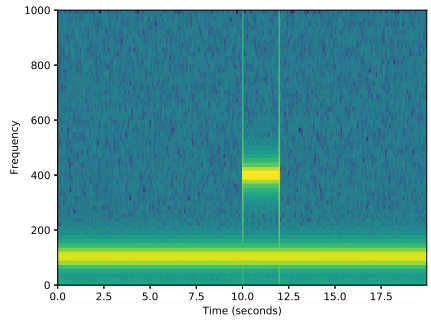
$$STFT(\tau, f) = \int_{-\infty}^{\infty} x(t)g(\tau - t)e^{-i2\pi ft}dt \tag{2.11}$$

where τ denote the position of the window in the time domain, f denote the frequency of which to extract the amplitude, i is the imaginary unit and $g(t)$ is the window length function.

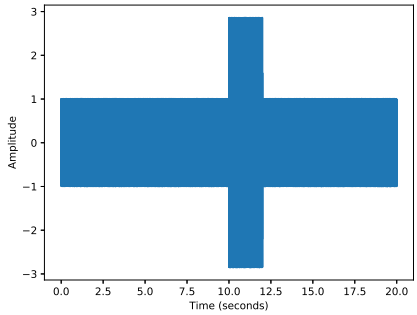
Only operating on a smaller segment of the signal reduces the dimensionality of the output frequency domain, and important information may be lost in this trade-off. One may



(a) The frequency space created by computing the Short Time Fourier Transform on the signal in 2.6c with a window length of 0.5 seconds.



(b) The frequency space created by computing the Short Time Fourier Transform on the signal in 2.6c with a window length of 0.05 seconds.



(c) A signal given by the function $\sin(100t)$ with an added signal of $2\sin(400t)$ in the interval $[10.0, 12.0]$.

Figure 2.6: Three plots showing the trade-off between long and short window length in Short Time Fourier Transform. A blue color indicates a low amplitude, while a more yellow color indicates a higher amplitude. Plot 2.6a has a long window length, which results in a higher resolution in the frequency dimension at the expense of a loss in precision in the time dimension. Plot 2.6b has a shorter window length, which results in a lower frequency resolution, but a higher precision in the time dimension [26].

address this issue by adjusting the window length, where a larger window will lead to a higher frequency resolution, but at the same time reduce the precision in the time dimension. Equivalently, a shorter window will lead to higher time precision, but at the cost of frequency resolution. This trade-off between time and frequency is illustrated in Figure 2.6.

Harmonics

The fundamental frequency of a signal is the frequency where the signal appears to repeat at a rate f, which is usually referred to as the first harmonic. The second, third and further harmonics are then defined as integer multiples of the fundamental frequency, e.g., $2f$, $3f$ and so on. These harmonics are captured by the Discrete Fourier Transform (DFT) where the assumption is that the given signal is periodic, and the length of the window used to calculate the discrete Fourier transform is equal to the period of the signal.

As the signal is sampled with time interval I equal to the period of the signal, the DFT captures all the harmonic frequencies up to the $\frac{N}{2}$th frequency, where N is the number of samples per period. These $\frac{N}{2}$ Fourier coefficients are all harmonic frequencies of the fundamental frequency. If the period of the signal is not equal to the length of the signal being sampled by the DFT, the output of the transformation will contain non-zero components for frequencies that are not an integer multiple of the fundamental frequency. These non-zero components are known as spectral leakage [27].

Ideally, a voltage signal in an AC power grid would be a perfect sinusoidal waveform of a single frequency. In practice this is not the case, due to the presence of nonlinear components that draw power disproportionately to the voltage source [28]. If the nonlinear components draw power in a symmetrical fashion above and below its centerline, the components will introduce odd integer harmonics into the signal, while asymmetrical components will introduce even harmonics. Most nonlinear components that are used in the electrical grid today are symmetrical around the centerline, which results in the presence of odd harmonic frequencies in the power grid. Even harmonic frequencies are mostly absent in healthy operating power lines [28].

Total Harmonic Distortion

Total Harmonic Distortion (THD) is a measure of the distortion in a signal with a given fundamental frequency. The formula for THD is given by

$$THD = \frac{\sqrt{\sum_{n=2}^{\infty} V_n^2}}{V_1} \tag{2.12}$$

where V_k is the RMS value of the nth harmonic component of the signal [29]. This measure compares the presence of the harmonic components to the fundamental frequency, and is often denoted as THD_F.

Another measure of THD, often referred to as THD_R, compares the RMS values of the harmonic components to the RMS value of the complete signal, not only its fundamental value. THD_R is given by

$$THD_R = \frac{\sqrt{\sum_{n=2}^{\infty} V_n^2}}{\sqrt{\sum_{n=1}^{\infty} V_n^2}} \tag{2.13}$$

where V_k is the RMS value of the nth harmonic component of the signal. This measure gives a relative measure between 0 and 1, measuring how distorted the signal is from its intended fundamental frequency [29].

Wavelet Transform

A wavelet is a rapidly-decaying, wave-like oscillation that has zero mean, and where the wave begins and ends with an amplitude of 0. An example of a wavelet function is illustrated in Figure 2.7.

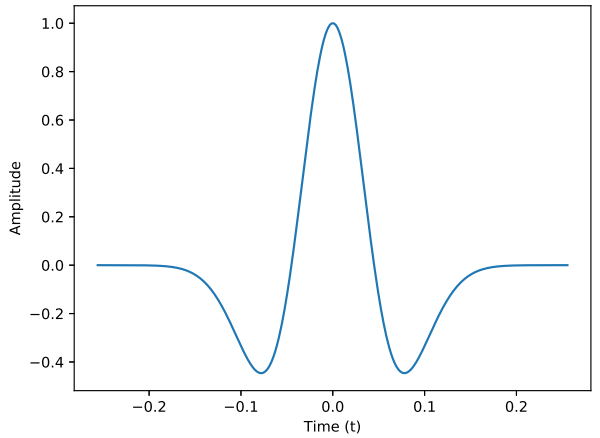


Figure 2.7: An example of a wavelet function. The function is given by $\frac{1-2\pi^2 f^2 t^2}{e^{\pi^2 f^2 t^2}}$ with $f = 5$ and referred to as the Ricker wavelet [30].

Wavelet Transform is often used in signal analysis. It is similar to the Fourier Transform in that it approximates a given signal by fitting a number of functions to the signal, but differs in which functions it uses for approximation. While the Fourier Transform approximates functions by fitting an infinite sum of sine-functions to the original signal, a Wavelet Transform uses wavelets to transform the signal from its time domain into the frequency domain.

The transformation uses a given wavelet, often called the analyzing function, and a varying window length, which is translated along the time dimension and scaled in size. This enables the transformation to capture frequency information at different window sizes, thus varying the time-frequency trade-off depending on the frequency. High-frequency

signals will get a high precision in time, while low-frequency signals will get a lower precision in time. The output of a Wavelet Transform is a 3-dimensional tensor describing the frequency space with time translation, scale and amplitude as its dimensions.

S-transform

The S-transform is a generalization of the Short Time Fourier Transform, where the Short Time Fourier Transform is modified to enable varying window sizes, to extract a frequency dependent resolution [31].

The following Gaussian function is used to define the window size:

$$g(t) = \frac{|f|}{\sqrt{2\pi}} e^{-\frac{t^2 f^2}{2}} \tag{2.14}$$

By replacing the window function $g(t)$ in 2.11, we get the following definition for the S-transform [31]:

$$S(\tau, f) = \int_{-\infty}^{\infty} x(t) \frac{|f|}{\sqrt{2\pi}} e^{-\frac{(\tau - t)^2 f^2}{2}} e^{-i2\pi f t} dt \tag{2.15}$$

As the relation between time window length and frequency is inversely proportional, the transform will use a wider window width on low frequencies, and narrower window width on high frequencies. This ensures a better time resolution for higher frequencies and better frequency resolution for lower frequencies.

2.1.3 Three phase power

In single-phase systems, instantaneous power dissipation is changing along with the amplitude of the sinusoidal signal, which results in an uneven power supply. To deal with this problem, one has devised a balanced three-phase system, where the instantaneous power is constant over time. A three-phase power generator is built by placing three coils $\frac{2}{3}\pi$ radians away from each other on a circle, with a rotating magnet in the center. This leads to three different voltage waves, equal in magnitude and frequency but out of phase from each other by $\frac{2}{3}\pi$ radians, with the same properties applying to the generated current. Figure 2.8 illustrates the offset in phase between the three waves. While there are multiple benefits to this system, it also leads to some more possible faults. If one of the phases go out of sync, it will lead to an unbalanced supply [32].

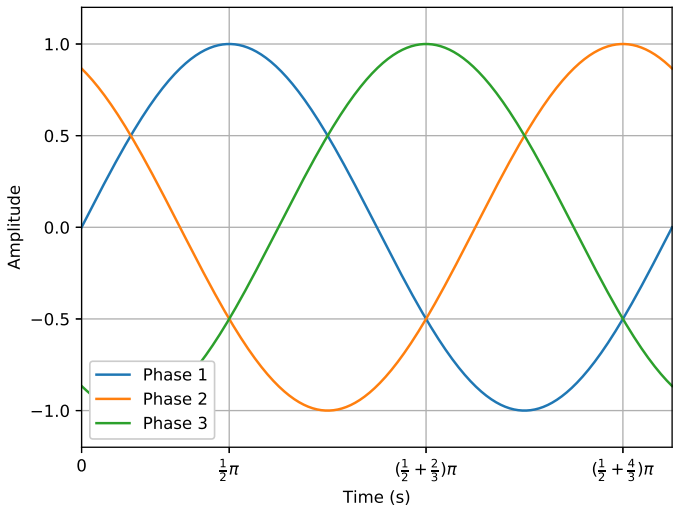


Figure 2.8: Three phase power. The three signals are phase offset by $\frac{2}{3}\pi$ from each other.

2.2 Faults and disturbances

In Norway 2017, a total of 895 disturbances occurred in the electrical power transmission grid. 32.4 % of these consisted of faults on power lines [5]. Of these again, nearly half led to interrupted delivery, meaning power was not delivered to customers, while 13 % caused an interruption duration of more than 30 minutes. On average, interruptions like these cause a socioeconomic cost of 800 million NOK each year, also known as the CENS cost [11]. In this section, we look at the reason for some of these faults, as well as the different types of voltage disturbances that occur in a power grid.

2.2.1 Cause of faults

In a complex power grid, there can be multiple reasons leading to a fault in the electrical network. A power line can go down because of weather, e.g., a tree falling on a line due to the wind, a thunder strike, or icing caused by cold weather. In Statnett's report from 2017 [10], they have mapped the causes of each fault in the Norwegian grid. The results are summarized in Table 2.1.

The surrounding environment alone amounts to 41.1 % of all faults in the grid for the years 2009-2017, while it is the reason behind 73.9 % of all undelivered power. The surroundings are further categorized into more fine-grained categories, which are listed in Table 2.2.

	Antall driftsforstyrrelser				ILE pga. driftsforstyrrelser			
	Antall		Andel		MWh		Andel	
Utløsende årsak (hovedgruppe)	2017	Årsgj.snitt 2009-2017	2017	Årsgj.snitt 2009-2017	2017	Årsgj.snitt 2009-2017	2017	Årsgj.snitt 2009-2017
Driftspåkjenninger	63	51	8,8 %	6,2 %	79	151	4,0 %	3,0 %
Konstr./montasje mm	62	57	8,7 %	7,0 %	42	136	2,1 %	2,7 %
Mennesker (andre)	11	11	1,5 %	1,3 %	34	54	1,7 %	1,1 %
Mennesker (personale/innleid)	74	89	10,4 %	11,0 %	72	107	3,7 %	2,1 %
Omgivelser	231	333	32,4 %	41,1 %	1 119	3 725	56,9 %	73,9 %
Teknisk utstyr	159	171	22,3 %	21,2 %	523	612	26,6 %	12,1 %
Tidligere feil	12	5	1,7 %	0,6 %	9	4	0,5 %	0,1 %
Årsak ikke klarlagt	102	93	14,3 %	11,5 %	88	252	4,5 %	5,0 %
Sum	**714**	**809**	**100 %**	**100 %**	**1 965**	**5 042**	**100 %**	**100 %**

Table 2.1: Overview of causes of faults in the Norwegian power grid in 2017 [10].

For failures on power lines, fraction of failures caused by surrounding is even higher, as can be seen in Figure 2.9. The surroundings cause nearly all faults that occur on lines in the Norwegian power grid, and the lines are often placed in areas with none nearby to fix potential faults. This shortage of on-site personnel makes it even more beneficial to be able to detect or predict faults ahead of time, so that personnel can be deployed before a potential power outage.

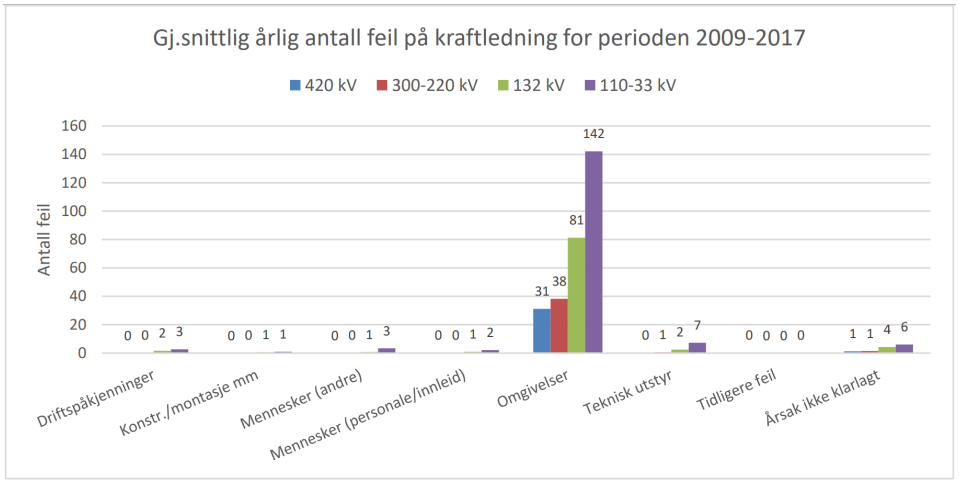


Figure 2.9: Faults on power lines in Norway categorized by cause [10].

We can see in Table 2.2 that animals, birds and lightning amount to 45 % of all faults related to surroundings in recent years. These causes are out of the scope of this report to predict. We are hypothesizing that other causes, namely vegetation, wind, snow, and

	Antall driftsforstyrrelser				ILE pga. driftsforstyrrelser			
	Antall		Andel		MWh		Andel	
Utløsende årsak:		Årsgj.snitt		Årsgj.snitt		Årsgj.snitt		Årsgj.snitt
Omgivelser	2017	2009-2017	2017	2009-2017	2017	2009-2017	2017	2009-2017
Fugl/dyr	2	7	0,9 %	2,2 %	2	9	0,2 %	0,2 %
Salt/forurensing	3	9	1,3 %	2,8 %	35	23	3,1 %	0,6 %
Snø/is	46	37	19,9 %	11,3 %	505	185	45,1 %	5,0 %
Tordenvær	60	129	26,0 %	38,8 %	239	232	21,4 %	6,2 %
Vegetasjon	34	45	14,7 %	13,5 %	160	541	14,3 %	14,5 %
Vind	51	75	22,1 %	22,4 %	122	2 592	10,9 %	69,6 %
Øvrige	27	21	11,7 %	6,3 %	54	115	4,9 %	3,1 %
Årsak ikke klarlagt	8	9	3,5 %	2,7 %	2	28	0,1 %	0,8 %
Sum	**231**	**333**	**100 %**	**100 %**	**1 119**	**3 725**	**100 %**	**100 %**

Table 2.2: Faults in the Norwegian power grid caused by surroundings in 2017 [10].

salt will lead to disturbances in the electric signal that will be detectable before a power failure occurs. A single failure may not be detectable in advance, but the cascading effects on other power lines might be detectable still, and further damage to equipment or loss of power can be avoided by taking the appropriate actions.

2.2.2 High impedance faults

High impedance fault (HIF) is a group of power system disturbances that happens when a conductor makes unwanted electrical contact with another surface element, for example a road, a tree or other vegetation. These faults restrict the flow of current by a level that is usually lower than what is reliably detectable by regular devices [33]. HIF is particularly dangerous since it is not only harmful to the electronic equipment but can also be dangerous to animals and people around it, as HIF can generate inflammable gases resulting in explosions or fires. The detection of these faults have therefore sparked an interest in the research community, and a number of algorithms have been applied to the problem[34]. The presence of high impedance faults can typically be detected when analyzing the waveform of a signal, with asymmetry and extra high- or low-frequency components added to the usual waveform [35].

2.2.3 Voltage disturbances

A power line can experience different disturbances which may eventually lead to a fault. In this section, a list of the most common known voltage disturbances is presented. Each disturbance is described as how it affects a sinusoidal wave signal and some of the possible causes. We will differ between the following 7 categories of disturbances based on

Seymour [36]:

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

Transients

Transients can be categorized into two types of faults: impulsive and oscillatory transients.

Impulsive transients are sudden peaks or surges in voltage level and can be the result of lightning or faults in the equipment used.

Oscillatory transients are changes in the steady-state of a voltage signal, typically causing an increase in voltage, and then a sudden loss, which causes the voltage level to fluctuate back and forth.

Transients are illustrated in Figure 2.10.

Interruptions

An interruption is the complete loss of voltage in the system and can be further categorized dependent on its duration. It is often caused by some damage to the line itself, e.g. from lightning strikes, animals, trees falling on the line, extreme weather or equipment failure. An interruption is easy to spot if it happens at home, as it typically causes all lights and electronic equipment to go black, only to come back shortly after. Voltage readings are useful for detecting interruptions, as the output will be 0 for a period of time, as illustrated in Figure 2.11.

Sag / Undervoltage

A sag is a reduction of the peak AC voltage, where the maximum amplitude drastically lowers for a few periods. This can occur during system faults or from heavy load machinery starting up in the power system. A sag with a duration longer than 1 minute is

classified as an undervoltage disturbance. The effect of sag on a signal is illustrated in Figure 2.12a

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

Figure 2.10: Example waveforms of transient disturbances [36].

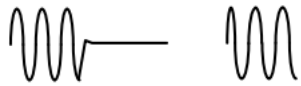


Figure 2.11: Example waveform of an interruption disturbance [36].

(a) Sag. **(b)** Swell.

Figure 2.12: Example waveforms of sag/swell disturbances [36].

Swell / Overvoltage

A swell is the opposite of a sag, with an increase in peak AC voltage for a given duration. If this duration is longer than 1 minute, it is called overvoltage. This often occurs as a result of load switching where the system is too weak to handle a needed voltage regulation. The effect of a sag on a signal is illustrated in Figure 2.12b

Waveform distortion

Waveform distortions are disturbances in the sinusoidal wave and have a variety of causes. The five most common distortions are:

1. DC offset
2. Harmonics
3. Interharmonics
4. Notching
5. Noise

DC offset DC offset is an offset in the sinusoidal wave such that the average value is not zero. This often causes unwanted current to devices that may already be operating at their maximum capacity and can cause overheating.

Harmonics Harmonics are corruptions in the sinusoidal wave at specific frequencies which are multiples of the fundamental frequency of the wave.

Interharmonics Interharmonics are waveform corruptions where a periodic signal which is not an integer multiple of the fundamental frequency of the signal is mixed with the original signal.

(a) DC offset. **(b)** Harmonics. **(c)** Interharmonics.

Figure 2.13: Example waveform distortions [36].

Notching Notching is a disturbance in the voltage level that is periodic in demand and could be seen as a periodic impulse problem, with constantly fluctuating voltage.

Noise Noise is an unwanted voltage or current imposed on the system from the outside and can be caused by poor grounding or other devices such as radio transmitters.

(a) Notching. **(b)** Noise.

Figure 2.14: Example waveform distortions (2) [36].

The different types of waveform distortions can be seen in Figures 2.13 and 2.14

Voltage fluctuations

Voltage fluctuation is a variation in the sinusoidal waveform that is systematic in its form, where the voltage differs between 95 % and 105% of its target voltage. This is typically due to a load on the system that has great variations in its demand. Voltage fluctuations are illustrated in Figure 2.15a

Frequency Variations

Frequency variations are the rarest type of problem occurring in an electrical grid, but is, as the name suggests, a variation of the frequency in the voltage, which can be seen in Figure 2.15b

(a) Voltage fluctuation. **(b)** Frequency Variations.

Figure 2.15: Example waveforms of frequency variations and voltage fluctuations [36].

Chapter 3

Background - Machine Learning

The contents of this chapter is based on a semester project conducted in collaboration with Vegard Hellem during autumn 2018. The project report is available online [12].

This chapter describes common machine learning and feature extraction methods used in statistical learning applications. The scope is limited to methods used in this project. The project report from fall 2018 contains further information about Wavelet transform and S-transform, two methods frequently used in signal analysis.

Section 3.1 starts off by introducing notation and and concepts used by multiple statistical methods within machine learning. Section 3.2 describes in detail the machine learning and feature extraction methods used in this thesis. Finally, Section 3.3 briefly describes some of the metrics commonly used in machine learning applications.

3.1 Machine learning

A machine learning algorithm is an algorithm that can **learn** from data. The definition of learning is widely discussed, but a commonly used definition by T. Mitchell is:

> 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 [37].

3.1.1 Notation

We will use the following notation throughout this chapter:

Function estimators will be denoted as the function name with an added hat, e.g. an estimator of the function F will be denoted as $\hat{F}$. Functions that are defined by a set of hyperparameters θ will be denoted as F_θ.

Vectors will have a bold font with lowercase letters, e.g. $\mathbf{x}$, while scalars will have a normal weighted font, x. Matrices will be written in uppercase, with a bold font, e.g. $\mathbf{X}$.

For series of variables, we will use the notation $x_{1:n}$ to denote the series of variables x_1, x_2, x_3 ... up to and including x_n.

3.1.2 Introduction to machine learning

A common goal for machine learning is to approximate a function $\hat{F}_\theta$ to an unknown function F, which takes as input a scalar or vector $\mathbf{x} \in \mathcal{X}$, where $\mathcal{X}$ is the domain of possible inputs, and outputs a scalar or vector $\mathbf{y}$. The goal is to maximize the performance of a model, with respect to some loss measure L. This is done by finding the parameters θ of the model, such that they minimize the loss L.

Problems in the machine learning domain are often divided into three categories; supervised learning, unsupervised learning and reinforcement learning. In this report we will focus on the problem category supervised learning, where a dataset of labeled samples is provided.

The fundamental assumption in supervised learning is that given enough training samples, the machine learning method can create a model that will be able to generalize to new, unseen inputs as well. In the instance of fault prediction in power grids, the goal of the machine learning model could be to predict the probability of whether or not a fault will occur within a limited time interval. The loss measure L can be defined as the difference in predicted probability of a fault and the ground-truth of the provided sample, whether or not a fault will occur within a specified time interval.

Moving back to the definition of learning, in the context of fault prediction in power grids, the task T is predicting faults, the experience E is the input of sensor data, while Performance Measure P is the value of the loss defined by L. A machine learning method would then be said to learn if the loss of the model decreases, given more samples of input sensor data.

Machine learning differs from some other Artificial Intelligence (AI) methods, in that feature extraction often is performed by the method itself. There is often no human intervention required for rule generation, as the method is capable of recognizing and extracting general patterns in the dataset.

Overfitting and underfitting

Two of the most well-known problems in machine learning are overfitting and underfitting. As the goal of supervised learning is to approximate an unknown function by using a dataset of samples, it is a common problem that the model either adapts too well to the input data or is unable to approximate the unknown function because of lack in model capacity. This is undesirable, as we want the model to learn the general patterns found in the input space, and not adapt too much to the noise in the data samples. If the model is unable to approximate the function due to lack of model capacity, we call it underfitting. If the model adapts too well to the training dataset, and ends up memorizing the data samples, we call it overfitting.

There are multiple ways to deal with overfitting and underfitting of models, which can roughly be divided into two categories: data augmentation and model tuning. The process of making the model more robust is called generalization, and methods from both categories are often used to reduce the risk of overfitting or underfitting.

The goal of generalization is to reduce the estimated generalization error, the model loss when presented with new, unseen samples. To estimate this error, one usually divides the dataset into three categories; a training set, validation set and test set. The training set is used as input samples when training the model, and the validation set is used to estimate the generalization error when faced with new samples. By tuning the hyperparameters of the model to achieve a best possible generalization error estimate on the validation set, we can thereafter test the model on the test set, to get a final generalization score. It is important not to use the test set during model training or hyperparameter tuning, as it will make the model conditionally dependent on the test data, and introduce a bias into the final generalization score. The same can be said about using the validation set as training samples during model fitting, as this will also introduce a bias in the generalization estimator.

Figure 3.1 shows the relationship between over-/underfitting and the generalization error when training a model with sufficient capacity on multiple iterations of the same dataset.

Data augmentation

By augmenting the samples in the dataset with label-preserving transforms, one is able to increase the size of the dataset without explicitly collecting new data samples. There is a multitude of transforms that are label-preserving in nature, but the label-preserving property is dependent on the problem at hand. An example is the problem of classifying images into a number of classes. Label-preserving transforms can then be translating the image, changing the lighting or rotating the image. The label will still be the same, as the image still expresses the same concept of an object.

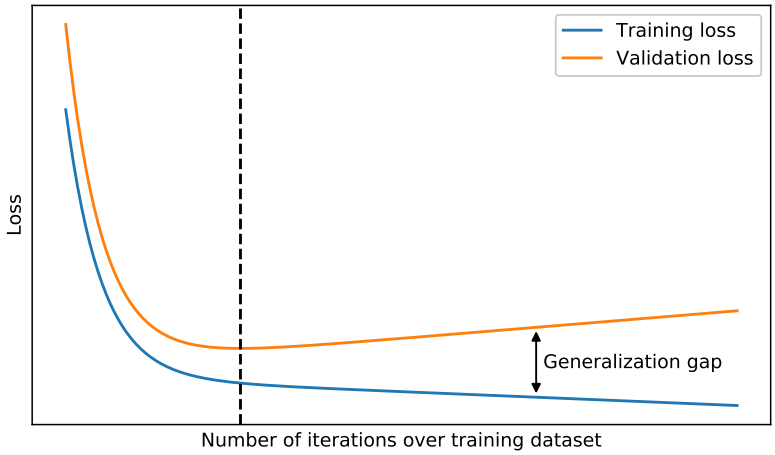


Figure 3.1: The relationship between over-/underfitting and the generalization error when training a model with sufficient capacity on multiple iterations of the same dataset. The training loss keeps decreasing while the validation loss reaches a global minima at the dashed line. After the dashed line, the model is said to overfit to the training data.

Model tuning

A machine learning model usually consists of an architecture defined by a set of hyperparameters. Learning rate, the number of layers, input weighting and a number of trainable parameters are only a few examples of the many parameters that might be tuned to create a better model.

By systematically testing all possible combinations of hyperparameters, it should in theory be possible to create the model that is best suited for the problem at hand. In practice, however, this approach is unfeasible for models with a large number of hyperparameters. The number of possible combinations increases exponentially with the number of parameters, and most machine learning models use considerable time to learn a new model given parameters and input data. This makes the approach too time-consuming for practical use.

In practice, one often seek simple models with few parameters because of interpretability and to avoid overfitting. By reducing the number of trainable parameters one often reduces the expressiveness of the model in function space, resulting in forcing the model to learn the actual underlying distribution of the data, instead of memorizing all samples in the training set. A good analogy is trying to fit a single line through 5 points in $\mathbb{R}^2$. A linear model might be too simple, while a 10th degree polynomial is guaranteed to fit a curve through all the points, but might lead to an overly complex function estimator, leading to a high generalization error.

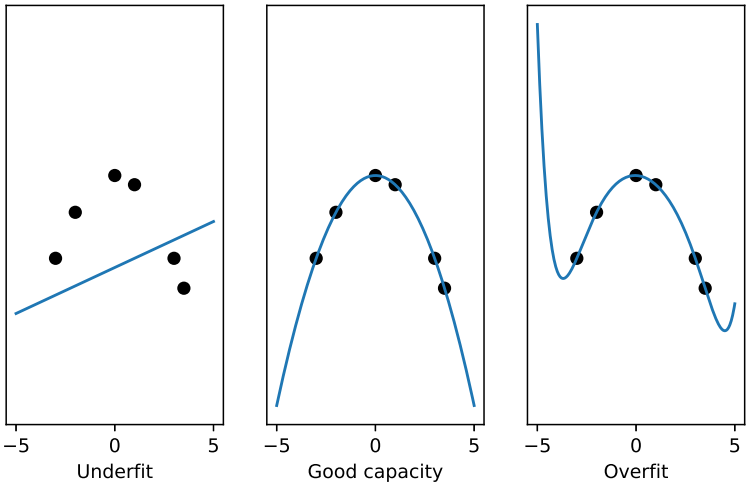


Figure 3.2: Model capacity related to underfitting and overfitting. A model with too little capacity might not be able to approximate the underlying distribution of the data. An overly complex model might overfit and generalize poorly to new data. [38]

Figure 3.2 illustrates the relationship between model capacity and approximated functions.

An approach to avoid overfitting that is often used in practice is regularization, a term covering multiple methods used to tune machine learning models. An example of regularization is adding a penalty to large weights as part of the loss function in the model's target function. Putting a penalty on the magnitude of weights in the model favours function approximations with sparse or low-valued weight matrices. This in turn helps increase the interpretability of the model, which is favourable, as a common goal in machine learning is to answer the question of why the model behaves the way it does, not just to know that it behaves correctly.

Two common regularization penalties are the L_1 and L_2 norm. The definitions are given below,

$$L_1 = \lambda \sum_{i=1}^{m} |w_i| \tag{3.1}$$

$$L_2 = \lambda \sum_{i=1}^{m} w_i^2 \tag{3.2}$$

where m is the total number of weights in the model, λ is a hyperparameter determining the weight of the loss term and w_i is a model weight. Increasing λ increases the penalty

for large weights, and will result in sparser models, but might lead to a too sparse model if set too high, as the model will prioritise to use small/few weights to reduce the loss rather than to actualyl predict the correct output.

Another common approach to reduce generalization error is a method called ensemble learning. The method consists of training an ensemble of multiple models rather than training a single model to perform prediction. Each individual model calculates an output given the input, and the final output is determined by aggregating all of the outputs.

There are multiple ways to train an ensemble of models, where Bagging [39] and Boosting [40] are two common algorithms. Bagging is the process of training multiple models on different subsets of the dataset, where all the outputs are combined by taking the mean value of the individual outputs. Boosting is an iterative process of model training, where the next model in the iterative process increases the loss contributed by previously misclassified samples, which creates a model that better predicts outliers in the output space.

3.2 Machine learning methods

In this section, we introduce specific machine learning and feature extraction methods that have been proven to be effective on the type of problems that arise in classification problems and dynamic systems. We will cover:

- Deep Learning
- Hidden Markov Models
- Decision Trees
- Support Vector Machines

3.2.1 Deep Learning

Neural networks have seen huge improvements and thus risen in popularity the recent years, mostly due to an increase in computation power [41]. It was initially inspired by the way the human brain works, and consists of layers of nodes and weights combined with a linear function, with an added non-linear function applied element-wise for each output of a layer. Deep Learning is often mentioned along with neural networks, as it is simply a subcategory of neural networks, specifically a network with multiple layers between the input and out layers. Such models are called deep neural networks [41] and is what we will cover in this section. Figure 3.3 illustrates the difference between a simple neural network and a deep neural network.

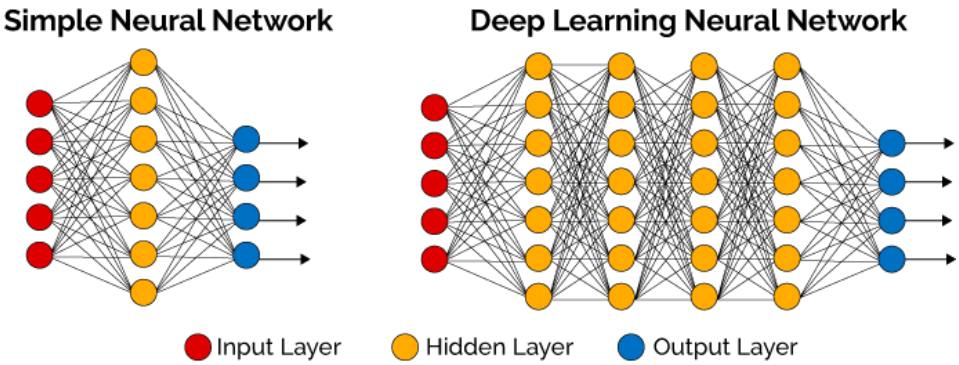


Figure 3.3: The difference between simple and deep neural networks. Deep networks contains more layers between the input and output layers than a simple neural networks [42].

A neural network, and consequently a deep neural network, consists of a number of layers which define the mapping

$$\mathbf{y} = \hat{f}_\theta(\mathbf{x}) \tag{3.3}$$

where $\mathbf{y}$ is the output of the network, $\mathbf{x}$ is the input, and θ is parameters of the network which are optimized to the best function approximation with respect to some loss measure L. For each layer l, each individual node x_i outputs the weighted sum of its inputs, with a non-linear function applied to the sum of inputs, defined as

$$x_i^l = \sigma(\sum_{j=1}^{N}(x_j^{l-1} \cdot \mathbf{W}_{j,i}^l) + b_i^l) \tag{3.4}$$

where x_i^l is the output of node i in layer l, N is the number of nodes in the previous layer, $\mathbf{W}_{j,i}^l$ is the weight from node j in the previous layer to node i in the current layer, b_i^l is an added bias term and σ is a non-linear function.

One of the more commonly used non-linear activation functions is the Rectified Linear Unit (ReLU), defined as:

$$f(x) = \max(0, x) \tag{3.5}$$

To optimize the performance of a neural network, a set of labeled samples in run through the network to generate predictions. The output of the network is then compared to the labels of the samples, which is regarded as the ground-truth label for each sample. A loss measure can then be calculated by the use of a comparison function, to calculate the total error of the network for the given samples. By using the gradient descent algorithm, the

weights in the network are adjusted to reduce the total loss by propagating the error layer-by-layer backwards through the network and calculating the individual contribution to the loss for each weight in the network [41].

Further details on how the gradient descent algorithm is implemented through backpropagation in a neural network is thoroughly described in the Deep Learning book by Goodfellow [41].

Deep Learning has been widely successful, achieving a high accuracy and beating the previous state-of-the-art methods on a multitude of problems. While it may take a long time to learn and update the weights of the network, it is quite fast in producing values once it is fully trained [38].

3.2.2 Hidden Markov Models

Hidden Markov Models (HMMs) are probability based machine learning methods for predicting latent variables in time series data. It is one of multiple algorithms within the class of Bayesian classifiers. Given a series of observable variables, we want to predict the state of a latent variable, on which the observable variable is conditionally dependent on.

We will use $\mathbf{y}_t$ to denote the vector of observable variables at time step t , and $\mathbf{x}_t$ to denote the vector of latent variables at the same time step. In literature, $\mathbf{x}_t$ and $\mathbf{y}_t$ is often referred to as the belief state and the evidence, respectively [43].

An important assumption in hidden Markov Models is that $\mathbf{x}_t$ satisfies the *Markov Property* [37], defined as

$$P(\mathbf{x}_t|\mathbf{x}_{1:t-1}) = P(\mathbf{x}_t|\mathbf{x}_{t-1}) \tag{3.6}$$

which means that the variable at time t is only conditionally dependent on the state of the variable at the previous time step, and not any other previous states. The observable variable $\mathbf{y}_t$ must also satisfy the following conditional dependency

$$P(\mathbf{y}_t|\mathbf{x}_{1:t}, \mathbf{y}_{1:t-1}) = P(\mathbf{y}_t|\mathbf{x}_t) \tag{3.7}$$

which means the observed variable $\mathbf{y}_t$ is only conditionally dependent on the state of the latent variables at the current time step [37]. A Markov Model must also be a stationary process, which is the property that the transition probabilities between states and the conditional probabilities for the observed variable $\mathbf{y}_t$ does not change over time.

Hidden Markov Models was one of the earliest algorithms to show great potential for fault detection in dynamic systems [44], and is visualized in Figure 3.4.

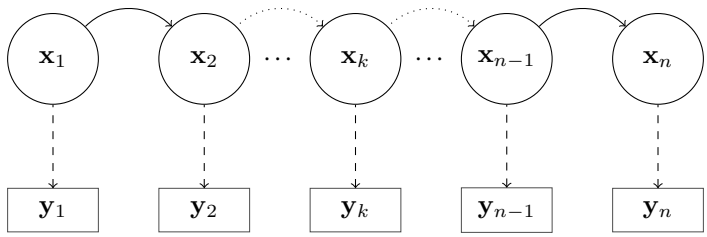


Figure 3.4: An illustration of a Hidden Markov Model where $\mathbf{x}_t$ is a latent variable, $\mathbf{y}_t$ is an observable variable, and arrows mark conditional dependencies between variables.

3.2.3 Decision Tree

Decision Tree is a machine learning method for classification problems that works by inferring rules for splitting the dataset into multiple subsets based on the properties of the data. It is similar to Expert Systems, except that the rules are created by using statistical properties rather than expert knowledge.

A Decision Tree creates rules for splitting by maximizing the *information gained* from performing a split. Information gained is defined as the reduction of Entropy in the set of samples, which for a binary classification problems is defined as

$$BinaryEntropy(S) = -\frac{|P|}{|S|} \cdot \log_2(\frac{|P|}{|S|}) - \frac{|N|}{|S|} \cdot \log_2(\frac{|N|}{|S|}) \tag{3.8}$$

where S is a collection of samples, P is the set of positively labeled samples in S and N is the set of negatively labeled samples in S. For Decision Trees, $0 \cdot \log_2(0)$ is defined as 0 [37].

For decision problems with more than two decision outcome values, we can generalize the definition of binary Entropy to multiple classes:

$$Entropy(S) = \sum_{i=1}^{C} -\frac{|S_i|}{|S|} \cdot \log_2(\frac{|S_i|}{|S|}) \tag{3.9}$$

where C is the number of decision outcome values, S is the set of samples and S_i is the set of samples in S belonging to class i.

The gain in information by splitting the dataset on attribute A can further be defined as

$$Gain(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \cdot Entropy(S_v) \tag{3.10}$$

where S is the set of samples, A is the attribute on which to split S and S_v is the set of samples where attribute A takes value v in S.

The attribute chosen to split each node into a number of subsets in the Decision Tree is thus defined as

$$\underset{a \in Attributes(S)}{\arg\max} \; Gain(S, a) \tag{3.11}$$

The naïve approach to make decisions with a Decision Tree is to keep splitting the dataset until the subsets are completely divided into sets of samples belonging to a single class. In practice this would lead to overfitting of the model, as the full set of samples might contain noise that would generate a lot of superfluous rules just to classify a few outliers or mislabeled samples. In order to counter this, model constraints in the form of maximum tree depth or a lower threshold for information gain can be set to stop the Decision Tree from splitting the samples into further subsets. The sample is then classified as the class which the most samples in the subset belongs to [45]. An illustration of a Decision Tree can be seen in Figure 3.5

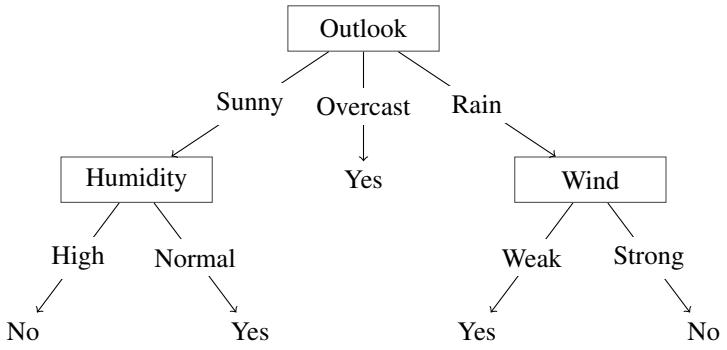


Figure 3.5: A Decision Tree for the binary decision problem of playing tennis or not. A decision is made by presenting a data sample with the properties $\{Outlook, Humidity, Wind\}$ and traversing the tree by picking the edge matching the value of the sample's property. Once a leaf node is reached, a decision has been made [37].

Decision Tree is a popular machine learning method as it is easy to explain the output of the model and how the rules are created, while at the same time providing good performance on multiple classification problems [46]. An extension to Decision Trees called Random Forest algorithm is an algorithm which produces an ensemble of Decision Trees

on different subsets of the total dataset and aggregates their outputs to get a more robust prediction model.

Random Forests can also provide a list of the attributes in the samples, ordered in terms of feature importance, which provides insight into which features are actually important for the classification problem at hand [47]. By calculating the mean information gained for each feature in all the Decision Trees in the Forest, the model is able to sort the features by importance.

3.2.4 Support Vector Machines

Support Vector Machine (SVM) is a machine learning method that has been widely used in classification problems. An SVM works by finding the hyperplane in a multidimensional space that separates the two classes that one wants to divide with the largest margin between the hyperplane and the nearest samples, often referred to as the *support vectors* [48]. A example is illustrated in Figure 3.6.

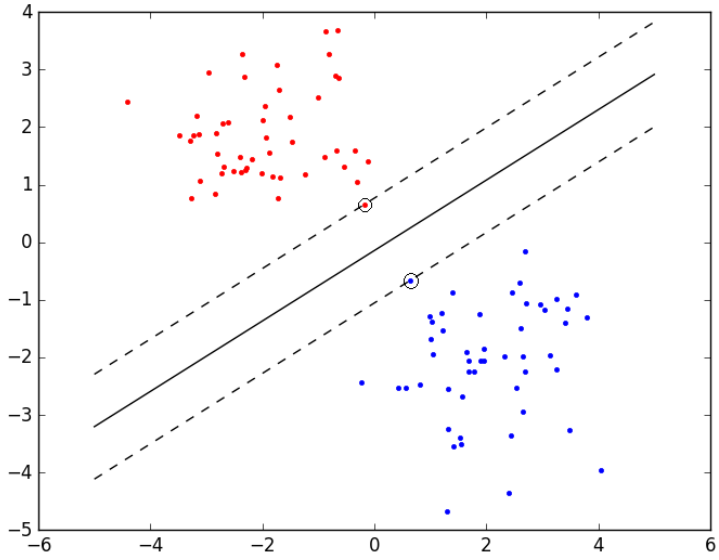


Figure 3.6: A Support Vector Machine separating a set of samples in $\mathbb{R}^2$ [49]. The hyperplane produced by the SVM is the hyperplane that maximizes the distance to the closest samples, popularly called *support vectors*. The support vectors are circled with a black border and lie on the dotted line.

The basic implementation of an SVM, often referred to as a *Hard Margin SVM*, requires the binary classification problem to be linearly separable. In practice, this is usually not the case, and the SVM provides multiple ways to deal with this scenario.

A first approach is to introduce a variable to count the number of misclassified samples, given a hyperplane. This transforms the SVM model objective from maximizing the margin to support vectors into a minimization problem of minimizing the number of misclassified samples, while maximizing the margin to the support vectors. This version of an SVM is known as a *Soft Margin SVM* [48].

Another approach is to transform the data points into another feature dimension where they are linearly separable. This method is often referred to as a kernel trick and a *Dual Form SVM*. By utilizing a kernel function, used to map the samples into a feature space where they are linearly separable, we can then classify the samples in this higher dimensional space into their respective categories before transforming the samples into their original feature space, but with an added classification label. Popular kernels are the polynomial kernels and the radial basis function / Gaussian kernel [50].

3.2.5 Machine learning in dynamic systems

> A dynamical system is a system whose state evolves with time over a state space according to a fixed rule [51].

The notion of time in dynamical systems adds complexity to many of the existing machine learning methods that were described in the previous sections. While most basic machine learning methods classify their data according to the current input, the added time dimension makes the output dependent on not only the *current state* of the data, but also the *change over time*. This adds another dimension to the input data, which can make the dimensionality of a single sample computationally intractable for classical methods.

Algorithms that work on time series data has to take into consideration the number of previous steps to look at by considering the value of more data versus the added time complexity of the algorithm itself. The more data to look at, the higher the complexity, and the more time is required to train and run the algorithm.

One of the more common usages of such sequential data involves Natural Language modelling, which is the task of predicting the next word, given a sequence of words [52]. Hidden Markov Models, as described in Section 3.2.2 is a popular way to deal with this, as is an extension to neural networks, which we describe in the next section.

3.2.6 Recurrent Neural Networks

One of the most popular ways to deal with sequential data is Recurrent Neural Networks (RNN). RNNs are neural networks that not only feed the output values forward to the next layer, but also uses the value as an input into itself in the next time step, as shown in Figure 3.7

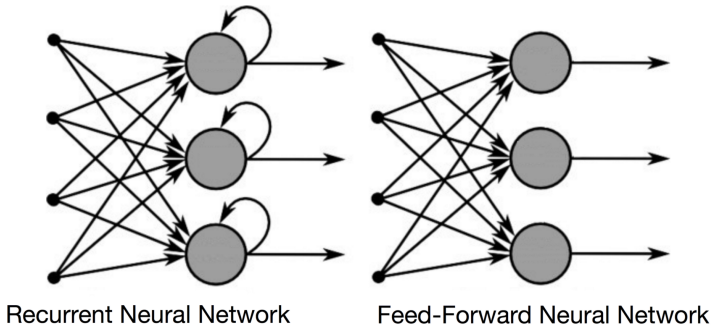


Figure 3.7: An illustration of the differences between a Recurrent Neural Network and a Feed Forward Network. A Recurrent Neural Network uses its output value as input in the next round of computations [53].

There are multiple implementations of RNNs where one of the implementations, called Long Short-Term Memory (LSTM), are responsible for much of the success in Deep Learning in areas where sequential data is used [53]. RNNs is one of the few models with internal memory, making it capable of remembering previously seen inputs from a sequence. This help increasing the accuracy of predicting what comes next, as RNNs can utilize use the entire sequence of state changes to predict the next value.

The LSTM implementation of RNNs make it capable to learn from experiences that may have a long time delay between a specific action and the resulting behaviour. This makes it especially useful for fault prediction in the power grids, where disturbances in the voltage signal sometime earlier could be the only telling of a fault that is going to happen. RNNs have previously proven successful in classifying voltage disturbances [54].

3.3 Metrics

There are multiple ways to compare the performance between machine learning models. In this section we will give a brief introduction to some of the common metrics used in classification and regressions problems.

3.3.1 Matthews Correlation Coefficient

Matthews Correlation Coefficient (MCC) is a measure of model performance in a binary classification problem [55]. It takes into account the true positive (TP) rate, true nega-

tive (TN) rate, false positive (FP) rate and false negative (FN) rate, and is regarded as a balanced measure, even for classification problems with an unbalanced dataset [56].

The MCC takes on values in the interval $[-1, 1]$, where a value of -1 means there is perfect negative correlation between the input variable and the dependent output variable, 0 means the two random variables are uncorrelated and 1 means there is perfect correlation. The MCC is given by:

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}} \tag{3.12}$$

3.3.2 Binary cross-entropy

Binary cross-entropy, also called logistic loss, is a measure of the similarity between an output probability and the ground truth of the output. It measures the number of bits required to encode the samples if an encoding is optimized for the output distribution rather than the true, underlying distribution. The

The Binary cross-entropy is given by [57]:

$$BCE(p, q) = -\sum_{x \in \mathcal{X}} p(x) \cdot \log q(x) \tag{3.13}$$

where p is the the true, underlying probability distribution of the samples in $\mathcal{X}$ and q is the distribution of the output samples from an estimator.

3.3.3 Receiver Operating Characteristic curves

A Receiver Operating Characteristic (ROC) curve is a plot showing the ability of an estimator to discriminate between true positive and false positive outcomes by changing the threshold needed for a classifying a sample as positive [58].

The Area Under the Curve (AUC) is the total area under the ROC curve, and is often used as a way to describe and compare ROC curves without drawing the actual curves for comparison. An AUC value of 1 means the model is able to perfectly classify as samples, without any false positives.

Chapter 4

Previous work

The contents of this chapter is based on a semester project conducted in collaboration with Vegard Hellem during autumn 2018. The project report is available online [12].

A lot of the previous work done in the field of voltage analysis has been done on artificially generated data and not on real data collected from sensors. There are multiple fields within power grid analysis, where a lot of the research effort on voltage faults has been on classification rather than prediction of faults ahead of time. Classification of faults is a natural first step on the road to predict faults, as classification is still done manually by operators after a fault occurs, and may provide insight into methods that are useful for prediction as well. Automation and improvements in this process would help in generating a large dataset for training supervised models for fault prediction.

Section 4.1 gives an introduction to the overall goal of this chapter, and introduces the different fields of study within voltage fault analysis. Section 4.2 covers research in voltage classification, a field of study close to voltage prediction, and Section 4.3 covers the field of voltage stability. Section 4.4 covers previous work within the field of voltage prediction, the area of research this thesis is a part of. Finally we give short summary in Section 4.5 that summarizes the findings across all fields of voltage fault analysis.

4.1 Goal of review

The goal of this chapter is to present and explain some of the background history and recent advancements in power fault prediction the last decades. We present the machine learning methods that have shown the most potential and the feature engineering and data types that have been used previously. To increase the area of research, we have looked not

only at the prediction of voltage faults but also at the steps coming after it, namely fault classification and stabilization.

One can roughly divide the timeline of voltage analysis into three separate intervals. The interval leading up to the fault event belongs to the field of voltage prediction, where the goal is to predict whether or not a fault will occur in the future. The interval where the fault event actually happened belongs to the field of fault classification. At this point, we are given a fault event and the goal is to classify it into which type of voltage fault actually occurred. The field also covers searching the whole timeline to detect faults that have not been manually labeled previously. The last interval is the field of voltage stability, where the goal is to determine whether the power line will stabilize or if it potentially will lead to an unstable network and a blackout. The separate intervals are illustrated in Figure 4.1.

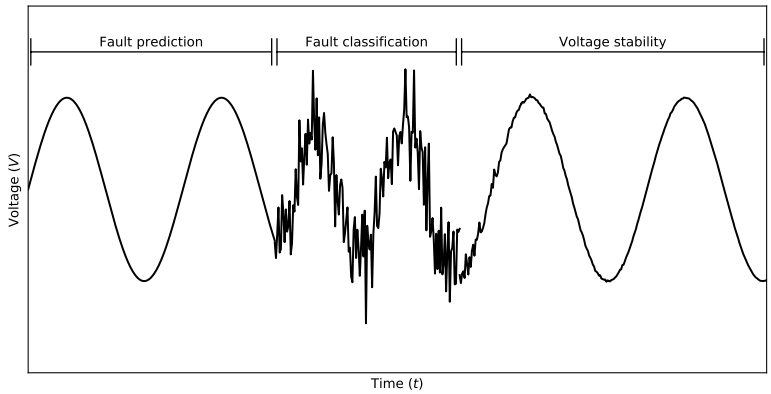


Figure 4.1: Timeline of the multiple intervals of interest in voltage fault analysis.

Two previous obstacles have been in the way of the development of power quality predictions; the lack of large-scale real sensor data and the lack of computation power. Both problems have partly been solved in recent years, with the deployment of Phasor Measurement Units (PMU) and Power Quality Analyzers (PQA) in the power grids and the evolution of processor performance and big data processing solutions. In Norway, sensors are deployed throughout the power grid and are continually monitoring important variables like voltage and current levels, sending sensor readings in real-time to a centralized server.

The problems of storage and computing power has been solved in the recent years, but processing the data with available machine learning methods is still a problem remaining to be fully solved. The sensor reading may be in a resolution as high as 50 000 records per second, while the system is still in need of real-time analysis. There is not much research on what sampling frequency that is needed to get a good result, but it is shown that some faults cannot be detected unless one has a sampling frequency of at least 25 KHz [59].

4.2 Classification of voltage disturbances

Since there are few publications on the prediction of voltage faults, we explore a similar field that has been more thoroughly researched over the years, namely the classification of voltage faults. While the classification of faults is not the overall goal of this report, a good classification method would be of immense use to a machine learning algorithm trying to predict faults, by using detected faults as training samples. Classification of voltage disturbances has been an important issue, as all classification of disturbances on the power grid normally is done through visual inspection of the disturbance by an operator. This procedure is susceptible to human errors, as well as being a very time-consuming practice [60].

Most research about classification is divided into two parts. The first part is feature engineering of the data, which involves transforming the data to reduce its dimensionality, while keeping as much as possible of the information it contains. This is done because monitoring data at a rate of 50 000 samples per second is still close to unfeasible for a real-time computer system, and machine learning systems might learn better from another representation of the data than a stream of voltage level readings. The second part is the processing itself, the classification algorithms used to obtain a result. Figure 4.2 illustrates the general steps of voltage classification.

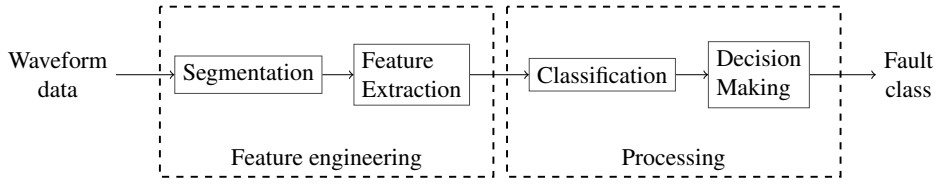


Figure 4.2: The two steps of voltage fault classification. Adapted from [61]

Some of the earliest work done on the classification of power disturbances using machine learning was done in 1995 by Ghosh and Lubkeman [62]. They describe a neural network approach for classification of waveforms, as part of a pre-processing step to collect data from transient recorders. They used sensor data that automatically triggered on certain measurement thresholds of disturbance and used this data to further classify the type of disturbance experienced. As seen in Section 2.2.3 most of the common voltage disturbances have a particular characteristic in their waveform. They achieved an accuracy of over 90 % with a neural network, but it should be noted that this classification was done on simulated data that had already been labeled by a set threshold and was thus made to classify disturbances that were already discovered.

Another early publication is a paper produced in 1999 on Power Quality Detection and Classification [63]. The authors used Wavelet Transforms to analyze problems both in time and frequency domains, which has been one of the most popular feature engineering tools used in power analysis and learning. By using a wavelet transform and calculat-

ing the standard deviation at different resolution levels of the signal, they classified voltage sag, swell, harmonic distortion and transient distortion. The classification was done through mathematical rules and definitions, and no machine learning method involving learning from the dataset was applied. However, the results of this study were unclear, and no benchmarked comparison to other solutions were performed. Although inevitably correct, as they used definitions of these interruptions for mathematical calculation, some disturbances that have unclear definitions or is very close to a disturbance could have been missed.

A review on different techniques and methodologies for disturbances classification was published in 2011 [64]. The traditional Power Quality Indices (PQI) includes peak values and total harmonic distortion, properties that have remained a standard in later work. The standard in feature engineering at that time for was Fast Fourier Transform, Goertzel's algorithm, Zoom FFT and Welch's algorithm among others. The information these algorithms provide is sometimes insufficient, and not all problems can be resolved by these algorithms, especially for detecting short spikes or transients [65]. The Short Time Fourier Transform fixes many of these issues as long as the window is short enough, along with the Wavelet Transform and S-transform.

For classification algorithms, one of the most widely used has been neural networks, while Fuzzy Logic also has shown great potential combined with expert-based systems [66].

Support Vector Machines (SVM) and their variants have also shown to be useful in the classification of voltage disturbances [67]. SVMs have been tested on data leading up to and including the actual disturbance and even done very well on data from other power line than it was trained on, leading to the theory that a pre-trained factory SVM could be deployed in multiple surveillance grids [68].

In 2013 a real-time power quality disturbance classification method based on extensive feature extraction using a hybrid of known methods by He, Li and Zhang was published [69]. To transform their input data they used a modified version of the S-transform. They argue that for real-time computation, many of the more computational heavy machine learning methods are undesirable, and instead use Decision Trees to classify the different type of disturbances. The system was tested on real-world data and performed well, with a $>$95 % accuracy on data sampled at a 52.2 kHz rate. Decision trees have later been successfully applied in multiple instances [70], also using the S-transform as a feature extraction method, with decision trees outperforming SVMs in a comparison done by Ray [71] in 2014.

In 2015, to deal with the problem of adapting to changes over time, Barros proposed an algorithm that could continuously learn from new input. The system is based on the wavelet transform and a neural network that was able to do continuous learning. Accuracy rates with continuous learning rose compared to a pre-trained network, going from an 83.51 % accuracy on classifying voltage swells to 100 % [60].

A paper using an LSTM deep learning approach for classifying voltage dips was published

Method	Advantage	Disadvantage
Fourier Transform	Simplicity of implementation.	Temporal information of frequencies is lost
Short Time Fourier Transform	Simplicity of implementation. Includes temporal information in Fourier Transform	Trade-offs between time and frequency resolution
Wavelet Transform	Time-frequency resolution	Computationally expensive. Some information regarding transients is lost
S-Transform	Phase correction	Block processing, may not be suitable for real-time requirement

Table 4.1: Feature extraction methods used for voltage fault classification. Adapted from [64].

in 2018. The method was strictly based on learning from training data, with no other inputs or definitions of voltage disturbances. Features were extracted from the raw waveform data by calculating the RMS value with a sliding window a single cycle at a time. The segments were fed into 4 LSTM layers, which extracted features before feeding the representation into a fully connected layer that classified the sample into one of seven classes. They achieved an accuracy of 93.5% on a dataset consisting of 5982 labeled voltage dips [72].

Feature extraction and machine learning methods for voltage fault classification is summarized in Table 4.1 and 4.2 with their mentioned advantages and disadvantages.

4.3 Voltage stability

Closely related to voltage fault prediction is the field of voltage stability. Voltage stability is a problem class where the goal is to detect whether any instability is present in the waveform of the voltage to reduce blackouts. This instability is usually a result of some of the faults mentioned in Section 2.2.3, gradually leading up to an interruption. The main difference between prediction and stability is that in stability the monitoring is done after an instability or fault has already happened. The task of monitoring the signal therefore has to look at a shorter interval, to see whether the power quality is stabilizing or worsening. The operator already knows that something is wrong, and wants to predict the consequences of the fault.

In 2009, Decision Trees were used to predict voltage stability following blackouts. Data from synchronized PMUs was used to predict whether the signal was stable or unstable. By using decision trees they were also able to find out which variables that were effectively

Method	Advantage	Disadvantage
Fuzzy Logic	Easy to analyze results. Outputs probabilities.	Requires expert knowledge to generate rules.
Neural Networks	Great flexibility in architectures. Can handle temporal relations in data.	Requires large, labeled datasets. High computational cost during training.
Support Vector Machines	Handles large feature spaces. Flexibility in choice of kernel and penalty term.	Requires large, labeled datasets.
Decision Trees	Simplicity of implementation. Computationally cheap.	Does not handle temporal relations in data.

Table 4.2: Promising algorithms used for voltage fault classification. Adapted from [64].

splitting the dataset. Classifying an insecure case as a secure case was made more costly than classifying a secure case as insecure, as the consequences from false positives would be more severe in a real system [73]. To deal with different types of data giving different values, they proposed extending the decision tree to a Random Forest algorithm to increase accuracy.

Random Forests have since been successfully implemented and tested by Guo and Milanović in 2013 when predicting if the disturbance would be critical for further system operations, achieving a 95 % accuracy on data from simulated PMUs 2.5 second after a fault. In 2015, Negnevitsky et al. also tested a periodically updated Random Forest classifier for on-line prediction, using a security index with degrees of altertness instead of a boolean output [74]. They achieved an accuracy of 99 % on a simulated 53-bus system of PMU data.

There have been multiple approaches with neural network architectures in monitoring voltage stability. In 2015, Zhukov proposed a hybrid architecture consisting of a fully-connected, a Kohonen and an Elman neural network, using data from a simulated power grid [75]. They achieved an accuracy of 97 % when predicting voltage margins following a fault.

A more recent paper was published in 2018, where Ibrahim and El-Amary proposed a recurrent neural network to predict voltage instability in power grids [76]. The project was done on simulated PMU data, on a standard 14-bus and 30-bus IEEE system using MATLAB for simulation. The network consisted of a fully-connected layer followed by a recurrent layer. The network was trained with the Particle Swarm Optimization (PSO) algorithm, which creates multiple networks with random weights and trains them in paral-

lel to reach the best known minima. They achieved an accuracy of 96.7 % and 97.5 % on 14-bus and 30-bus systems when predicting one of three labels *stable*, *alarm* and *trip*.

Guo and Milanović [77] proposed decision trees as a method of classifying the state of a system after a fault has occurred and whether it was a critical fault or if the system could keep operating. They achieved a 95 % prediction rate

4.4 Voltage disturbance prediction

Little work has previously been done in the field of fault prediction. The prediction process is very similar to the classification process as seen in Figure 4.2, however, the classification step is now a prediction of future faults that might occur and not a classification of an already detected fault. Since the work is done on the same data, we work under the assumption that the same methods of segmentation and feature extraction that are used in the fault classification and voltage stability fields can be used for prediction. We also assume that the methods used in classification can be adapted to solve this problem. One notable difference is that voltage prediction might have to look at at a larger interval of data than the classification and stability processes and is thus more dependent on handling long time series with temporal relations than many classification methods.

Some of the earliest work seen on voltage fault prediction was done in 1999 on a simulated 97-bus system modelling the National Power Grid of England and Wales [78]. Voltage sags were predicted using stochastic processes, generating voltage dips in simulated phasor data. This was a purely probabilistic prediction based on earlier fault events, time between faults and grid topology. The goal was to predict the frequency of voltage sags for a given site based on historical sensor data.

In 2016, researchers from Enerq in collaboration with the University of Sao Paulo developed a computer system to detect high impedance faults (HIF) in distribution lines [79]. The system used Power Quality Analyzers installed at feeders with a sample rate of 512 samples per 60Hz cycle to generate high frequency resolution data. The data was compressed and stored in a PQSCADA database provided by ElSpec. The four following algorithms were used for feature engineering:

1. Short Time Fourier Transform to extract harmonics and sequence components in the current signal
2. Wavelet transform to analyse disturbances in currents
3. Sum of differences Algorithm, adapted from [80]
4. Threshold detection of short time RMS value elevations

The system consisted of a two-step alarm system, where a neural network with a single hidden layer and the threshold detection system would trigger a stage 1 alarm that warns

the distribution grid operator. The alarm would further trigger another detection block, which could escalate the alarm to a stage 2 alarm.

An interesting find from the feature engineering process is that only a few harmonic components were needed to extract useful information about the currents. Only the 2nd, 12th and 14th harmonic components were compared to a threshold calculated from the average value of the harmonic values in the signal and used in the detection system. The detection system architecture and the two-step alarm system is illustrated in Figure 4.3 and 4.4 respectively.

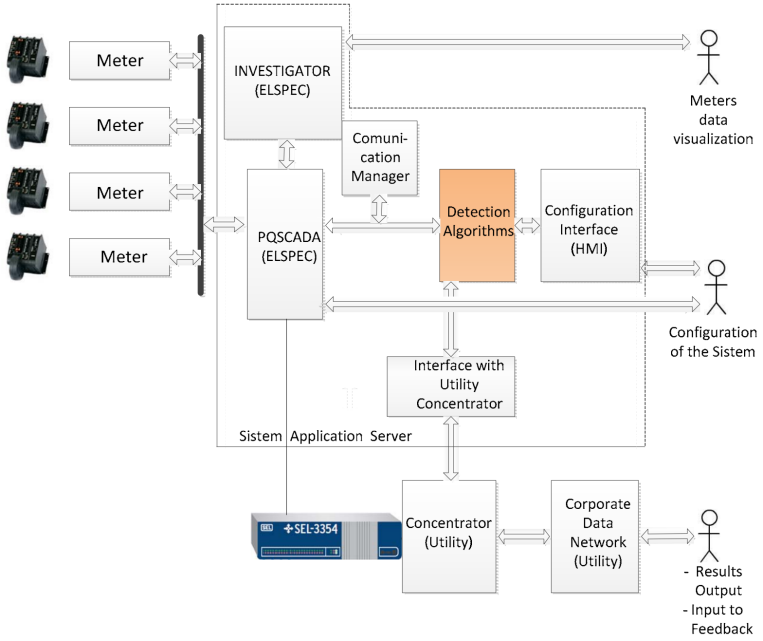


Figure 4.3: System architecture used by researchers from Enerq and the University of Sao Paulo to detect high impedance faults in the distribution grid [79].

Another high impedance fault prediction system was proposed by German researches in 2013, with the goal of detecting trees falling on power lines [81]. The system was tested on both simulated and on actual field measurements from PMU sensors, with a manually created algorithm for detecting an abnormal change in impedance due to disturbances on the power line. The system proved successful at detecting HIFs up to about $10^6\Omega$.

Early prediction of voltage sags caused by large rotor swings was explored by Weckesser et al. in 2014 using simulated data from PMUs in generators [82]. They used three different approaches, all based on statistical analysis of the voltage signal and threshold values. While they were able to predict the voltage falling below a critical value ahead of time, their definition of ahead of time was 300 millisecond before the fault. While this may be enough for a computer system response, a manual operator would have no chance of

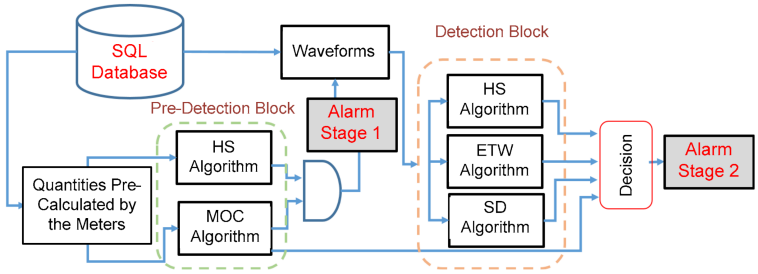


Figure 4.4: A two-step alarm system used by researchers from Enerq and the University of Sao Paulo to detect high impedance faults in the distribution grid [79].

responding to such a prediction.

The Texas A&M's Distribution Fault Anticipation project has since 1990 been trying to anticipate faults in power lines through real-time intelligent monitoring [83]. In their paper from 2009 describing their efforts, they showed numerous cases of power outages that had incipient faults leading up to the outage as far as one month before.

In the following report the year after [84], they showed that intelligent algorithms could detect some of these cases. The researchers behind the paper state that voltage harmonics at a sufficient sample rate is the the most important metric when considering widespread deployment of this system. We note that no testing was done on whether these algorithms would incorrectly detect disturbances in signals without a fault as well.

The same researchers from Texas A&M published a paper proposing a method to find recurrent disturbances in power lines using a clustering algorithm in 2017 [85]. The algorithm could signify a more extensive outage happening shortly due to failing or degrading line apparatus.

In 2018, Xiao and Qian used a Hidden Markov Model to predict power quality disturbances, using real data from a Chinese city and its local distribution grids [86]. The data consisted of PQA measurements with a sample rate of 256 samples per cycle, paired with the weather conditions at the sensor placement sites, along with 8010 labeled faults in the time series data. Their choice of model was influenced by their motivation to be able to derive the relations that exist in non-stationary time series. The model was successful in predicting power faults varying between 1 and 20 days before the fault event, with an accuracy close to 80 % in the most promising regions.

4.5 Summary

By combining feature extraction methods and machine learning models from different fields within voltage fault analysis, it should be possible to develop models aimed at fault

prediction. In this section, we summarize the findings from the fields of fault classification, voltage stability and fault prediction that show potential applicability to further fault prediction systems.

Most of the referenced studies in this report have used time-synchronized data from PMU sensors as their data source. As the data made available by SINTEF in this report is PQA data with a higher sampling frequency than PMU sensors, we hypothesize that the same methods used on PMU data can be successfully combined with the dataset available. Some extra preprocessing of the high-frequency data may be required in order to make the models computationally tractable.

4.5.1 Feature extraction methods

The feature extraction methods that have shown the most potential are Wavelet transform, a modified S-transform and Short Time Fourier Transform. These methods have already been successfully applied in systems showing good results on real-world data for fault classification, which proves that they can extract information from the signal about its current state. If there are disturbances in the signal before a fault occurs, we hypothesize that these extraction methods should be able to pick up these disturbances. The system created in collaboration with Enerq specifically used the wavelet and STFT for feature engineering, which further proves this hypothesis.

4.5.2 Machine learning models

Decision trees, SVMs and deep neural networks show the most potential for choice of model for processing dynamic power quality data. Both deep recurrent neural networks using an LSTM as cell architecture and more basic feed-forward networks have shown to be successful for classification and stability monitoring, in combination with the listed feature extraction methods. Random Forests and SVMs have also shown great potential for fault classification. Out of the few studies on voltage disturbance prediction, neural networks have also shown to be successful in this field, at least for high impedance faults.

Other models that have been tested in practice are fuzzy logic and Bayesian classifiers, where Hidden Markov Models have have been used in fault prediction on real data. In Table 4.3 we summarize our conclusions about what has shown promising results, and what we suggest exploring further.

Machine learning method	**Explore further**	**Justification**
Decision Trees	Yes	Promising results on classification, can extract rules to see how it has come to its conclusion.
Neural Networks	Yes	Been used in fault prediction systems on real data, LSTM can handle time series data very well.
Support Vector Machines	Yes	Performs well on high-dimensional data, has shown good results on voltage classification in multiple power lines.
Fuzzy Logic	No	Requires expert domain knowledge.
Bayesian Classifiers	Yes	HMM handles time series data. Been used successfully in predicting faults on real data.

Table 4.3: Machine learning methods that have shown promising results in previous work.

Chapter 5

Predicting faults in the Norwegian Power Grid

The contents of this chapter is partly based on a semester project conducted in collaboration with Vegard Hellem during autumn 2018. The project report is available online [12].

We start by presenting the EarlyWarn project in Section 5.1, a project by SINTEF that aims to predict power failures in the Norwegian power grid. We then continue in Section 5.2 by describing the sensors and data that is available in the power grid of Norway, which is used in this thesis. Lastly, we look at how the results from this thesis and the EarlyWarn project can be used in an application for power grid operators in Section 5.3.

5.1 EarlyWarn

EarlyWarn is a collaboration project between SINTEF Energi, NTNU, Statnett and multiple power grid operators in Norway. Their ultimate goal is proactive detection and early warning of incipient power grid faults and instabilities by using data from sensors placed at strategic places in the Norwegian power grid [1]. In combination with the recent advances within the field of machine learning, they hope to create a system which learns the early patterns of the voltage disturbances described in Section 2.2.3. The system can then be deployed as an application for continuous monitoring of the power grid at operation centers around Norway.

The project involves the transmission, regional and distribution grid levels, which is mon-

itored by sensors that store their records at centralized servers, operated by the grid companies. Through the use of Big Data processing solutions and machine learning methods, the project will bring the following value propositions to the involved parties:

- Improved situational awareness in control centres
- Targeted and efficient handling of faults and instabilities
- Improved understanding of the value of sensor data in a digital world
- Insight into required utility in future generations of instrumentation
- Improved competence in analysis and use of power quality sensor data

The monitoring application should provide a warning horizon long enough so that a human grid operator has time to react, inspect further details about the potentially upcoming fault and to take preventative measures to reduce the impact of the fault. This could be to dispatch a team of engineers for physical inspection and testing of equipment on the site of potential failure, or to shut down the power line completely, routing the power through other parts of the grid without a blackout. There is no hard limit on how long time is required to react to a potential power failure, but a rule-of-thumb is that the longer time, the better.

5.2 PQA data in Norway

Power Quality Analyzers (PQAs) are installed at strategic locations in the Norwegian power grid and give real-time insights into the power quality at these sites. These are one of the main type of sensors used by grid operators in Norway to monitor the health of the power grid at all times. An image of a PQA sensor is shown in Figure 5.1.

In addition to PQA sensors, Norway also has Phasor Measurement Units (PMUs) installed in the power grid. The disadvantage of using PMU data compared to PQA data is that PQA data has a superior sampling rate, and are thus able to obtain more detailed data on power quality [59]. PQAs has a bandwidth of 25 kHz versus PMUs' 50 Hz. This makes it possible to detect signals and noise that would otherwise have gone unnoticed due to the gaps between each sample. An example of this is seen in Figure 5.2 where the distortion in the signal is obvious when looking at the waveform with a high sampling rate, but inconspicuous at a lower sampling rate.

The added resolution in PQA data gives makes it possible to extract valuable information on different voltage quality parameters, including a high number of harmonic components, transients, and local voltage variation.

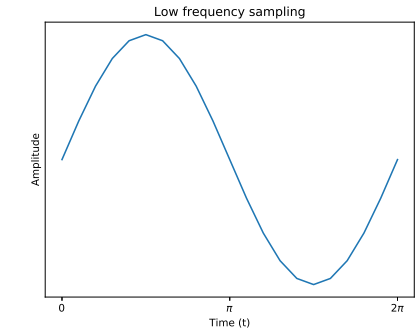
Figure 5.1: An example of an Elspec Power Quality Analyzer [87].

The drawback of PQAs compared to PMUs is that they are not time-synchronized with each other, which makes it harder to compare data from different sensors to aid in pinpointing the location of faults. PMUs also have the ability to stream measurement data live from the grid, while PQAs have a processing delay before the data can be transferred.

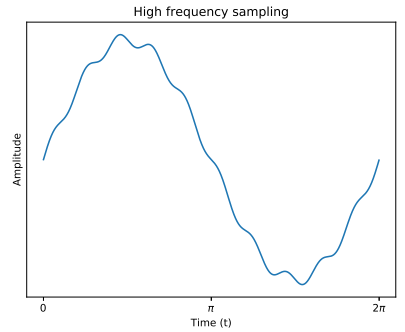
PQA data is collected by power distribution system operators (DSO) and tranmission system operators (TSO) in Norway. As part of the EarlyWarn project, SINTEF has received access to some of the data for research purposes. An important aspect to remember is that in an actual application for fault prediction, the data needs to be analyzed in real-time or with the shortest delay possible to have any value in the prediction of faults. Figure 5.3 shows the locations of built and planned PQA sensors in the Norwegian power grid by Statnett in 2016.

5.2.1 Available data

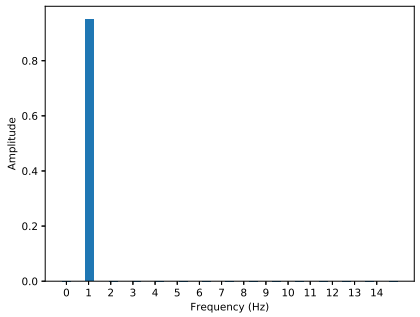
To enable machine learning methods based on supervised learning, we require a dataset of labeled fault events. The time series data is already made available to SINTEF through



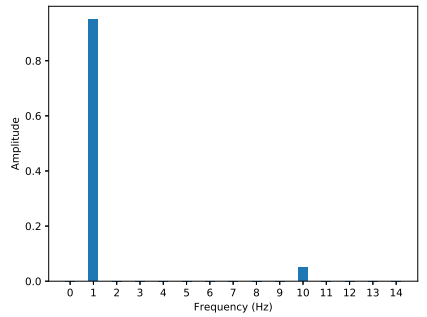
(a) A signal sampled at 20Hz per cycle.



(b) A signal sampled at 2000Hz per cycle.



(c) The Fourier coefficients for the signal in (**a**).



(d) The Fourier coefficients for the signal in (**b**).

Figure 5.2: Two graphs showing the same signal given by $0.95 \sin(t \cdot 2\pi) + 0.05 \sin(10 \cdot t \cdot 2\pi)$, sampled with low and high frequency. The signal sampled at 20Hz (**a**) does not capture the noise added in the signal, and only has a single Fourier coefficient. The signal sampled at 2000Hz (**b**) manages to capture the noise in the signal, which shows up as a second Fourier coefficient.

EarlyWarn, while the labels must be provided elsewhere. For this, SINTEF has created an application for automatic detection of power system faults, called AHA (Automatisk Hendelsesanalyse) [20]. AHA is able to classify the following type of faults:

- Voltage sags
- Power interruptions
- Ground faults
- Rapid voltage changes

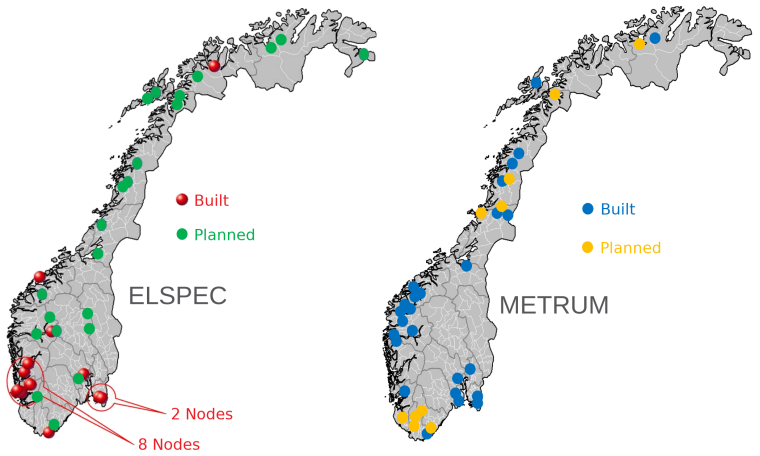


Figure 5.3: Geographical locations of built and planned PQA sensors from Statnett in 2016 [88].

The application is also able to distinguish between real and false voltage sags, which are caused by saturation in voltage transformers.

The software works by providing it with a start and end timestamp, which it uses to search the database of voltage time series, outputting a list of fault event timestamps in the given time period, where the fault happened and which type of fault the event is classified as. The list of timestamps can then be used further to extract time series data to create a suitable training dataset.

Further postprocessing of the fault event timestamp list can be done due to the Norwegian regulation on system responsibility in power grids [89]. The regulation instruct all power grid operators in Norway to analyze, document and report faults that occur in the grid. By cross-checking the list of generated timestamps against the list of reported faults from the grid operators, one can filter out false positives from the list of timestamps. However, there might also be cases where AHA has correctly detected a fault, which might not have been reported by the grid companies.

The current standard for fault reporting in Norway is the FASIT system [90]. The reports are generated by grid operators and uploaded to Statnett through a web-based interface. The extent of reporting can be imperfect, and the grid operators are only reporting power interruption faults through the system. Also taking into account the possibility of fault events that have not previously been detected by grid operators, the job of filtering out false positives from the list of timestamps is a tedious and time-consuming job, which has to be performed manually by experts in the field. Because of this, postprocessing of the list of timestamps has only been done to a small degree.

5.2.2 Dynamic Dataset Generator

To extract time series data belonging to the events in the list of timestamps, SINTEF has created an application to automatically extract the time series in question from a centralised database containing the data shared by collaborating system operators. The application is named Dynamic Dataset Generator, and provides both a graphical and command-line based interface to users. An image of the graphical user interface can be seen in Figure 5.4.

Figure 5.4: The graphical user interface of Dynamic Dataset Generator (DDG).

DDG lets the user specify a range of parameters that are used when extracting data from the time series database. The full list of tunable attributes are found in Table 5.1.

The database contains time series for multiple parameters, namely *Voltage*, *Current*, *Active power* and *Reactive power*. In this thesis we will only use voltage measurements in our experiments. For each of the mentioned measurements, multiple computed values can be extracted for each sample in the time series. All computed values are aggregated by the *Min*, *Max* or *Average* operator for the specified interval set by the Resolution parameter. The list of possible values to extract is found in Table 5.2.

Each of the computed values can be extracted for each line or phase. As the electrical grid in Norway is a three-phase system, there are three phases (Phase 1, 2 and 3) and three line (Line 1-2, 2-3 and 1-3) voltages available.

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

Table 5.1: Table of parameters for Dynamic Dataset Generator.

Computed value	Description
RMS	The RMS value of the signal, computed cycle-by-cycle in the original signal.
Waveform	The sampled amplitude of the original signal.
Harmonics	The harmonics computed cycle-by-cycle by the Discrete Fourier Transform, up to and including the 512^{th} harmonic.

Table 5.2: Table of available values to extract when generating a dataset.

In order to have a balanced dataset for supervised learning, a set of samples that do not contain fault events is required. DDG supports generation of such events, where it can generate a user-defined ratio of fault-to-non-fault samples. The non-fault generation algorithm works by taking a fault sample which belongs to a specific operation node, then randomly choosing an interval equal to the length of the fault sample within the operational interval of the node. If the randomly chosen interval overlaps with the fault event interval (including transient period after the fault), another interval is randomly chosen until the two intervals don't overlap.

By generating non-fault samples based on fault events, we ensure that all nodes are represented in the non-faulty dataset. As there are multiple labeled faults for each node during the operational interval, there is a non-zero probability that the randomly chosen interval overlaps with one of the other labeled fault events for the given node. However, the probability of this happening can be ignored in practice, as the operational period for each node spans across multiple years, where each node usually only contains around 300 labeled faults, and the extracted intervals are usually not more than a few hours at most.

The final output of DDG is a dataset of samples for each fault in the list of timestamps and an equivalent number of non-fault samples that can be used for further feature engineering

and machine learning. Figure 5.5 illustrates the entire workflow required for the data generation process. The dataset is saved as a comma separated list of values (.csv), with added metadata for each sample. The metadata included in each sample is described in Table 5.3.

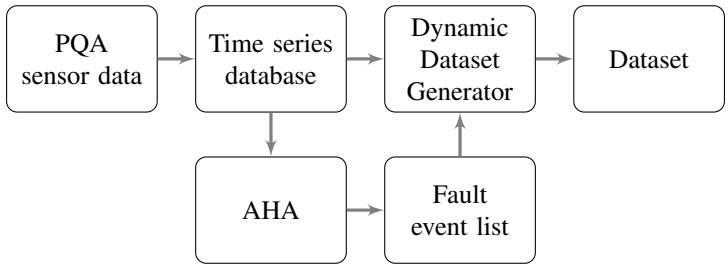


Figure 5.5: The process of generating a dataset of labeled faults and non-faults.

5.3 Fault prediction as a practical application

5.3.1 Real-time requirements

In a system where the goal is to warn power grid operators of potential faults or disturbances in the voltage signal, real-time data processing is of utmost importance. The warning horizon might not be more than a few minutes for many critical faults, and there are multiple processing steps that has to be completed before the operators are even made aware of an impending fault, which they then have to evaluate and act on.

The first step of the process is for the PQA sensors to record and send the sampled signal to centralized servers. The current setup which is used for research purposes contains a single centralized database that stores time series for multiple power grid operators. This might not be the scenario in the future, where each operator might have to operate their own database.

The sensors placed in the power grid has to provide a continuous stream of data to the database, to be further processed in the pipeline. Newer sensors [91] have the capability of processing the signal to extract features on-site, which helps reduce the load of the database servers. However, in the case of older equipment without these capabilities, the infrastructure has to be robust enough to handle both processing of input data and processing of aggregated values from the database. An important aspect is that the sensors also have to be connected to the Internet, as manual gathering of signal data will be too slow to have any value in a real-time monitoring system.

The second step of the process is to use feature extraction and machine learning methods to process the most recent signals captured by the sensors. For this step, two aspects have

Name	Description
Fault Detection	Whether the data contains a fault or not
Fault type	What type of fault it is, if any
Fault time	When the fault occurred
Start time	Start time of the first sensor reading
End time	End time of the last sensor reading
Total duration seconds	How many seconds of data the dataset contains
Total duration days	How many days of data the dataset contains
Resolution ms	The interval between each sampled data point
Time buffer seconds	How many seconds of data from the period after the fault occurred is included in the sample
Time transient seconds	The minimum amount of seconds that should pass between the end of a faulty sample and a non-faulty sample
N points	The number of data points for each parameter
Node	The name of the node from which the sensor data is accessed
Nominal voltage	The line voltage of the equipment at the fault location

Table 5.3: Metadata included per sample in the dataset.

to be considered when designing a system with real-time requirements: prediction time consumption and online model training.

The time used for feature extraction and model prediction is greatly influenced by the choice of algorithms and the length of time series data used in prediction. Storing the most recent time series data obtained from the database locally on the machine performing prediction can greatly increase the performance, as the machine only has to request the data that has been recorded since last time in regular intervals. Depending on the feature extraction method, it might also be possible to use extracted features from previous calculations, if the features does not depend on the complete time series, but only a local interval.

The system should also be able to learn continuously from new faults that occur in the power grid. Some machine learning methods use considerable time to train before they reach an adequate performance level, and taking the monitoring system offline while training such models is unacceptable. Luckily, it is possible to parallel the process of prediction with an older model and training of a new model. Seeing as the model performance probably won't change drastically by adding a few training samples to its training set, using an older model for prediction while training is an acceptable solution.

The third step is to incorporate the new system for fault predictions into the existing workflow of grid operators. There are already alarm mechanisms in place that warns of mal-

functioning power lines, and the added system introduces yet another set of alarms. It is important to consider the impact on attention and trust in the system, so that the operators can continue with an effective workflow. An important aspect to take into account is the risk of false positives and false negatives, which is covered in detail in Section 5.3.2.

The power grid operators holds the responsibility of making a system that fulfills the requirements of the first step of the process, and is out of scope for this thesis. Interaction designers should be involved in the process of incorporating the warning system into the grid operators' workflow, and is also out of scope for this thesis. The second step however, has to be taken into account when choosing the methods for feature extraction and machine learning. Results from architectures that require a lot of processing time for each sample might help us along the way to create a system for fault prediction, but for practical applications the end result has to be an application designed for real-time processing.

5.3.2 False positives and false negatives

When it comes to predicting a problem like power system faults, it is essential to take into consideration the consequences of false positives and false negatives. We will give a quick explanation of both in the context of fault prediction.

A false positive error would be to predict a time series sample as an upcoming fault, even when the correct answer is that no fault will happen.

A false negative error would be to predict a time series sample with an upcoming fault as a non-fault, not advising any action for the operator.

The consequence of a false positive error is a reduced trust in the systems and wasted time for the operators, as long as they have ways to manually investigate the signal and find out that no action needs to be taken. The consequence of a false negative, however, might lead to damage to the power grid, causing a potential blackout and a more significant cost to society [92].

One might argue that a potential blackout is a more severe consequence than wasted time for operators, but one has to keep in mind that the prediction system comes as an addition to the already existing monitoring infrastructure currently in place. Not installing the system would lead to the the same costs that we already have today. Thus, any correctly predicted fault is an added bonus to the current system, while all false positives comes as an added cost in the form of extra work for the operators.

Ensuring that a prediction system has no false positives is easy - one simply has to never predict any faults. Ensuring that a prediction system correctly predicts all upcoming faults is also easy - one can constantly output an alarm. Both of these extremes would of course not provide a very good or reliable prediction system.

Finding a good balance in fault prediction precision is hard, as we want the system to

predict as many true positives as possible without reducing the trust in the system among the operators. A too sensitive system would lead to the scenario depicted in the story of The Boy Who Cried Wolf [93], where lack of trust in the alarms would lead to operators ignoring the warnings. There must be a balance between precision and recall, which must be taken into consideration when designing and testing the application for real world use.

5.3.3 Other data sources

While the scope of this thesis only includes analysis of voltage signals to predict upcoming faults, one might consider taking additional data sources into the machine learning models for prediction. Weather data may be used in addition to other measurements to provide better predictions, as the presence of rain, wind and snow has shown to be correlated with the occurrence of power line faults [94]. Investigating the potentially added effect of further data sources should be considered after an initial prototype has been developed.

Chapter 6

Experiments and Results

This chapter presents the experiments performed during this thesis. In total 6 different experiments were performed. In Section 6.1 the multiple datasets used in the experiments are presented. Section 6.2 gives the background and motivation for each question, and describes which experiments helps answering which questions. Section 6.3 describes in detail how the experiments were performed, and the results from each experiment are presented in Section 6.4.

6.1 Datasets

The dataset used in this thesis was labeled by SINTEF's AHA system, presented in Section 5.2.1 and generated by DDG, presented in Section 5.2.2. There is a total of 4499 detected fault events available, collected from 15 different locations, with voltage levels ranging from 15 kV to 300 kV. The total list of faults has the following number of faults in each category:

- 2308 Voltage sags
- 1786 Ground faults
- 270 Power interruptions
- 135 Rapid voltage changes

Multiple properties can be extracted from the dataset available. The full list of available properties is listed in Table 5.1 and Table 5.2. Some of the experiments use different

properties, and thus multiple datasets have been extracted. The datasets are named and given in Table 6.1, Table 6.2 and Table 6.3 for reference throughout this chapter.

Dataset 1	
Extraction parameter	**Value**
Total duration	3600 seconds
Resolution	1 second
Buffer	0
Transient	3600 seconds
Aggregation method	Mean
Extracted feature	**Specificity**
Single phase harmonics	Phase 1, 2, 3. 1st to 16th harmonic
Line harmonics	Line 12, 23, 31. 1st to 16th harmonic
Fault types	**Successfully extracted**
Voltage sags	2178
Ground faults	1730
Power interruptions	220
Rapid voltage changes	132

Table 6.1: Features extracted in Dataset 1.

Due to some missing time series data, not all fault event time series are available in the database. The number of successfully extracted fault samples is therefore included in the dataset tables.

When generating a dataset of faults, a dataset of non-faults of the same size as the list of faults is generated. As not all faults could to be generated, the total number of non-fault samples sometimes exceed the number of fault samples. In these cases, a number of samples are removed from the non-faults so that the datasets are of equal size. The removed samples were chosen uniformly at random.

For each experiment, a test set was extracted from the dataset used. The test set consisted of 20% of the available data, picked uniformly at random. The remaining 80% of the dataset will be referred to as the training data.

6.2 Experimental Plan

The research questions defined in Section 1.2 is repeated here for reference.

RQ 1. Can machine learning methods predict upcoming faults on power lines by ana-

Dataset 2	
Extraction parameter	**Value**
Total duration	900 seconds
Resolution	1 second
Buffer	0
Transient	3600 seconds
Aggregation method	Mean
Extracted feature	**Specificity**
Single phase harmonics	Phase 1, 2, 3. 1st to 256th harmonic
Fault types	**Successfully extracted**
Voltage sags	2236
Ground faults	1749
Power interruptions	229
Rapid voltage changes	133

Table 6.2: Features extracted in Dataset 2.

lyzing voltage measurement data?

RQ 2. What attributes in a voltage signal are suited for predicting faults on power lines?

RQ 3. What is the performance of some machine learning algorithms at predicting faults on power lines?

RQ 4. Are some types of faults easier to predict than others?

We will address each question in turn and provide the background and goal for each question.

6.2.1 Can machine learning methods predict upcoming faults on power lines by analyzing voltage measurement data?

The fundamental question that EarlyWarn needs to answer in order to successfully build a system to predict power faults is whether it is possible to predict faults at all. Without a confirmation on this question, further work on building a system for fault prediction will not be useful for the parties involved in the project. As mentioned in Section 4.4, similar predictions have been done previously, which is promising for our initial experiment.

Dataset 3	
Extraction parameter	**Value**
Total duration	3600 seconds
Resolution	1 second
Buffer	0
Transient	3600 seconds
Aggregation method	Minimum, Maximum
Extracted feature	**Specificity**
Single phase harmonics	Phase 1, 2, 3. 1st to 16th harmonic
Line harmonics	Line 12, 23, 31. 1st to 16th harmonic
Fault types	**Successfully extracted**
Voltage sags	2178
Ground faults	1710
Power interruptions	220
Rapid voltage changes	132

Table 6.3: Features extracted in Dataset 3.

In Experiment 1, Dataset 1 will be used in combination with variations of Support Vector Machines, Random Forest Ensembles and Feed-forward Neural Networks to determine if it is possible to predict faults with a 10 minute prediction horizon. Prediction accuracy and Matthews Correlation Coefficient (MCC) will be used as metrics to determine if the machine learning methods are able to predict upcoming faults.

In a balanced dataset with two classes, the probability of predicting the correct class for a given sample is 0.50. If any model is able to perform significantly better than random guess on the unseen data samples in the test set, we will have proven that it is possible to predict upcoming faults in power lines with a 10 minute prediction horizon.

6.2.2 What attributes in a voltage signal are suited for predicting faults on power lines?

Feature extraction methods are important for many machine learning algorithms to perform well on classification tasks.

The available time series can produce frequency components up to the 512^{th} harmonic for each step in the time series, but there is little research on which of these available parameters that are essential for fault prediction. Based on discussions and tests in the EarlyWarn project group, it was discovered that a lot of the harmonic frequencies in the

extracted datasets contained only zero values for the majority of the extracted time series. Having 512 harmonic frequencies for each line and phase results in 3072 features per time step, which can make some machine learning methods too slow for real-time use, while the empty frequency columns provide no added information. Reducing the number of dimensions to a minimum of dimensions that are actually useful when discriminating between faults and non-faults will be beneficial for the runtime complexity of both feature extraction and prediction, and might also provide valuable insight into which features in a voltage signal that are important parameters when analysing the signal for human operators.

To answer this Research Question, two experiments will be performed. Experiment 2 will use Dataset 2, and the presence of the harmonic frequencies in faults and non-fault samples will be examined. This will provide insight into which harmonic frequencies that are actually present in the two classes.

Through discussions and tests performed internally by other members of the EarlyWarn project, it became apparent that there were other features than the mean value of harmonics that should be used when predicting faults. Using the minimum and maximum values of the harmonics frequencies within a window of 1 second gave better results when used in combination with SVMs, Random Forests and Neural Networks.

Experiment 3 will use Dataset 3 and a Random Forest to rank the importance of each individual feature. This experiment will provide further insight into which features that are important to predict faults.

6.2.3 What is the performance of some machine learning algorithms at predicting faults on power lines?

By combining the results from the three first experiments, we should be able to create models that perform better than the initial models tested in Experiment 1. In order to answer this Research Question, we will in Experiment 4 benchmark the same machine learning models used in Experiment 1 in addition to a Recurrent Neural Network on Dataset 3. The models' hyperparameters will be tuned using a validation set before testing the performance on the test set. Accuracy, MCC and Receiver Operating Characteristic (ROC) curves will be provided for all models.

6.2.4 Are some types of faults easier to predict than others?

Different type of faults might have different causes. Some of these causes might not produce distortions in the voltage signal beforehand, while others might have large impacts on the voltage signal leading up to the coming fault event.

By examining which type of faults our models manage to predict, we might uncover that

some types are near impossible to predict, while other types have good potential for detection. Focusing on easy-to-predict faults might lead to quicker adaption of a system that is able to detect some faults in the grid. Focus on harder-to-predict faults might lead to other approaches in feature engineering, or to the ultimate conclusion that some faults do not have a characteristic distortion in the voltage signal leading up to the event.

To answer this Research Question, we will perform two experiments. In Experiment 5 we will train the best performing machine learning models from Experiment 4 on the full training dataset, then do prediction on the test dataset. By separating the predicted test samples into the four fault categories available, we will see if the model is better at predicting some type of faults than others. Prediction accuracy within each fault category will be used as measure.

In Experiment 6 we will train the best performing machine learning model from Experiment 4 on datasets made up of fault events of the same category, and predict on a test set of the same fault category. This experiment will show if machine learning models focusing on a single type of fault are better at detecting specific faults, or if the type of fault have little impact on overall performance.

6.3 Experimental Setup

The setup of each experiment is divided into two sequential steps:

1. Preprocessing
2. Model tuning and validation

Some of the experiments depend on other previous experiment results in order to be performed. An overview of the experiment dependencies and their individual outcome is given in Figure 6.1.

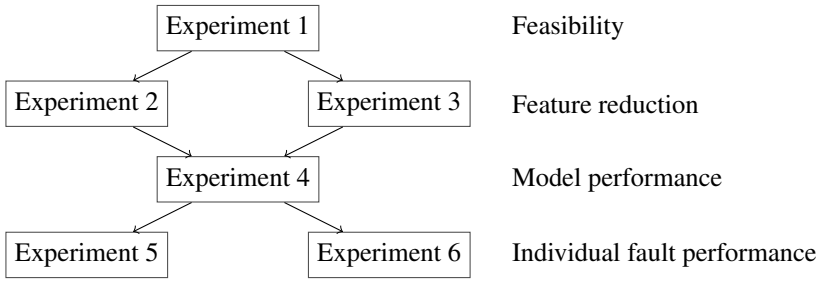


Figure 6.1: Overview of experiments with their outcome and dependencies on other experiments.

6.3.1 Experiment 1 - Feasibility test

Preprocessing

Dataset 1 was used in this experiment. In order to perform prediction rather than voltage classification, the last 10 minutes of time series data leading up to a potential fault was removed from each sequence, resulting in 50 minutes of data available in each sample.

The machine learning models used in this experiment do not handle time series data. In order to remove the temporal dimension, each sequence of harmonic frequencies in the sample was reduced by aggregating multiple functions along the time axis. The aggregation methods used were mean value, minimum value, maximum value and standard deviation. The resulting dataset contained samples of 386 features each, without any temporal dimension.

Model tuning and validation

As the intention of this initial experiment is to simply determine if it is possible to predict faults, and not to explore all possible preprocessing and machine learning models for the best possible prediction system, a few standard machine learning models were chosen. The machine learning models of choice are listed with their hyperparameters in Table 6.4. To implement the SVM and Random Forest models, the popular Python library Scikit-Learn v0.20.0 was used [95]. Neural networks were implemented in the Keras framework v2.2.4 [96].

Machine learning model	**Parameters**
Support Vector Machine	Kernel = Radial basis function
Random Forest	Number of trees = 200 Maximum depth = No limit
Feed-forward Neural Network	First layer size = 64 Second layer size = 32 Third layer size = 16 Optimizer = Adam Loss function = Binary cross-entropy Batch size = 128 Number of epochs = 150

Table 6.4: Machine learning models and the hyperparameters used in Experiment 1.

No models were tweaked to perform better on a validation or test set, and thus no validation set was generated for this experiment. Each model was trained on 80% of the total

available dataset and validated on the remaining 20% of the dataset. The neural network was trained for 150 epochs, where the MCC was calculated on the test set after each epoch. The results from the experiment is summarized in Section 6.4.1.

6.3.2 Experiment 2 - Harmonic frequency presence

Preprocessing

Dataset 2 was used in this experiment. In order to not include the harmonic frequencies that may appear during a fault, the last 5 minutes of time series data was removed from each sample, resulting in 10 minutes of data available in each sample. The final dataset was made up of samples with 600 time steps, each with 768 features per step.

The dataset was aggregated along the time dimension, counting the number of non-zero values in the harmonic feature series for each harmonic frequency. All labeled fault events were then summed together, creating a total sum of all non-zero values.

Model tuning and validation

The result was a count of the number of non-zero values for each harmonic frequency in the dataset, counted separately for faults and non-faults. The counts were divided by the total number of time steps in all the available samples in each category. The results are summarized in Section 6.4.2.

6.3.3 Experiment 3 - Feature importance

Preprocessing

Dataset 3 was used in this experiment. Similarly to Experiment 1, the last 10 minutes of each sequence was removed, resulting in a dataset of samples with 50 minutes of data.

As Random Forests do not handle time series data, the same preprocessing method used in Experiment 1 was applied to the features in Dataset 3. This resulted in a dataset where each sample consisted of 768 features, with no temporal dimension.

Model tuning and validation

In order to rank the feature importance of the available features in the dataset, a Random Forest containing 200 trees and no limit on tree depth was trained on the training data. The

25 most important features ordered by the reduction in information entropy by splitting the feature space is presented in Table 6.6.

6.3.4 Experiment 4 - Model performance test

Preprocessing

Dataset 3 was used in this experiment. All tested models except for the Recurrent Neural Network (RNN) used the same preprocessing method used in Experiment 1 and Experiment 3. Using the results from Experiment 2, all feature columns with less than 1% of non-zero values were removed from all samples. Using the results from Experiment 3, the most common feature aggregation methods used in the top 100 ranked features were applied to the remaining harmonic feature columns. This resulted in a dataset with 162 features per sample, with no temporal dimension. The complete list of features included in the dataset is listed in Appendix C.

For the RNN model, the same feature extraction and preprocessing methods used for the other models were applied, with the modification that the time series aggregation was not done on the entire time series at once, but rather in intervals of 100 seconds at a time. This resulted in a dataset of samples where each sample consisted of a time series of 30 steps, with 162 features per time step.

Model tuning and validation

Grid search was performed on each machine learning model to find the optimal hyperparameters. A validation set consisting of 20% of the training data was extracted to estimate the performance of each model before training the model with the highest MCC in each model category on the full training dataset. The test set was then used to give a final score to the chosen models.

The metrics recorded in each experiment were prediction accuracy, MCC, Receiver Operating Characteristic (ROC) curves and Area under The Curve (AUC). The final scores on the test set for the chosen models are presented in Table 6.7.

Support Vector Machines

Grid search was performed to find the optimal kernel. The kernels tested were linear, 2-4th degree polynomial, sigmoid and radial basis function (RBF).

Random Forests

Grid Search was performed to find the optimal number of trees and maximum tree depth. The number of trees to generate was tested between 50 and 500 at intervals of 50. The

maximum tree depth was chosen among { 5, 10, 15, 20, 25, No limit }.

Feed-forward Neural networks

A neural network with four fully-connected layers was used. The three first layers used Rectified Linear Unit (ReLU) as element-wise activation function, while the last layer used a sigmoid function with a single scalar as output. Grid Search was performed to determine the number of nodes in the three first layers.

Each model was trained for 300 epochs, where the model was scored using the validation set after each epoch. If the MCC score on the validation set improved, the weights of the network were saved. The highest MCC score achieved during training was used to rank the performance of the network.

Recurrent Neural Network

A Recurrent neural network was created to utilize the temporal relations in the input data. The model consisted of two sequential layers of bidirectional LSTMs outputting sequences, with an attention layer followed by two fully-connected layers. The two bidirectional layers had an output dimensionality of 128 and 64 dimensions respectively. The attention layer and the first fully-connected layer both had 64 dimensions in their output space, with the fully-connected layer using ReLU as activation function. The last fully-connected layer had a single scalar output with sigmoid as activation function. The network architecture was inspired by a Kaggle competition [97].

6.3.5 Experiment 5 - Fault category performance

Preprocessing

Dataset 3 was used, with the same preprocessing as described in Experiment 4. Each sample was marked with its fault category.

Model tuning and validation

The model with the highest MCC from Experiment 4 was chosen for this experiment. The model was trained on the full training set, and tested on the test set. The percentage of correctly predicted samples within each category are presented in Table 6.8.

6.3.6 Experiment 6 - Fault category specialization

Preprocessing

Dataset 3 was used in this experiment, with the same preprocessing methods as described in Experiment 4 and 5. The full dataset was divided into the four fault categories available, where a number of non-faults equal to the number of faults in each category were picked uniformly at random. 20% of the samples within each category were extracted for test sets, picked uniformly at random.

Model tuning and validation

The model with the highest MCC from Experiment 4 was chosen for this experiment. A model was trained for each fault category, using the training data consisting of samples from that particular fault type. The model was tested on the test set of the respective fault category, where prediction accuracy and MCC was recorded. The performance within each fault category are presented in Section 6.4.6.

6.4 Experiment Results

6.4.1 Experiment 1 Results

Table 6.5 shows the metrics recorded for the three models tested in Experiment 1. Out of the three models tested, the Random Forest model achieved both the highest accuracy and MCC, while the SVM achieved the lowest score in both metrics.

Model	**Accuracy**	**MCC**
SVM	0.5746	0.1791
Random Forest	0.6643	0.3329
Feed-forward Neural Network	0.6340	0.2748

Table 6.5: Accuracy and MCC scores for models tested in Experiment 1.

6.4.2 Experiment 2 Results

Figure 6.2 shows the percentage of non-zero values in the harmonic frequency time series for fault and non-fault samples respectively. A percentage of 100% would mean that a non-zero value is present in all time steps in all three phases in all samples in its respective

category. The figure only includes harmonic frequencies where they have non-zero values in at least 1% of the time steps for either faults or non-faults. A full list of the 256 available harmonic frequencies is available in Appendix A.

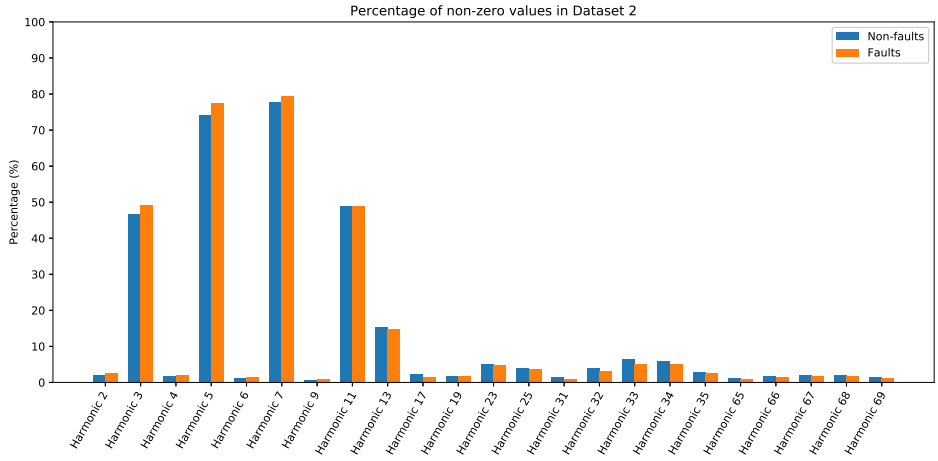


Figure 6.2: The percentage of non-zero values in the sequences of harmonic frequencies in Dataset 2.

Out of the 256 harmonics available, only 23 frequencies had more than 1% of non-zero values. Harmonic number 9 only appear more than 1% in the fault dataset, and have less than 1% appearance in the non-fault dataset. Harmonic number 65 appear more than 1% in the non-fault dataset, but have less than 1% appearance in the fault dataset.

Harmonic frequency 1, the fundamental frequency of the signal, has no appearance in the dataset. This is due to the fact that the harmonic frequencies retrieved from the database are normalized as a percentage of the fundamental frequency. Including the fundamental frequency in the dataset would only have given a column of 1's, and it has thus been removed from the database query.

6.4.3 Experiment 3 Results

Table 6.6 presents the 25 most important features according to the Random Forest feature ranking. The top 120 features are included in Appendix B.

The importance column contains the information gain calculated for each feature, and sums to 1 when summing all importance scores in the full feature set. A higher score is better, meaning that the feature provides a higher information gain when used as split attribute in the Decision Tree. As there are 768 features available, the information gain would be 0.0013 if all features were of equal importance. From the results, we can see that the most important features have around 5 times higher importance score than the default

Harmonic number	Time step feature	Line / Phase	Aggregation method	Importance
Harmonic 10	Minimum	V2	Maximum	0.00660
Harmonic 4	Minimum	V1	Standard deviation	0.00618
Harmonic 5	Minimum	V1	Standard deviation	0.00600
Harmonic 10	Minimum	V23	Maximum	0.00597
Harmonic 11	Minimum	V12	Standard deviation	0.00584
Harmonic 10	Minimum	V31	Maximum	0.00570
Harmonic 10	Minimum	V3	Maximum	0.00567
Harmonic 10	Minimum	V1	Maximum	0.00567
Harmonic 12	Minimum	V2	Maximum	0.00565
Harmonic 4	Minimum	V3	Standard deviation	0.00561
Harmonic 10	Minimum	V12	Standard deviation	0.00560
Harmonic 10	Minimum	V12	Maximum	0.00560
Harmonic 14	Minimum	V3	Standard deviation	0.00540
Harmonic 4	Minimum	V12	Standard deviation	0.00535
Harmonic 4	Minimum	V2	Standard deviation	0.00534
Harmonic 14	Minimum	V1	Standard deviation	0.00530
Harmonic 10	Minimum	V12	Mean	0.00529
Harmonic 15	Minimum	V12	Standard deviation	0.00528
Harmonic 15	Minimum	V23	Standard deviation	0.00527
Harmonic 15	Minimum	V3	Standard deviation	0.00525
Harmonic 4	Minimum	V31	Standard deviation	0.00523
Harmonic 11	Minimum	V23	Maximum	0.00519
Harmonic 14	Minimum	V23	Standard deviation	0.00519
Harmonic 5	Minimum	V31	Standard deviation	0.00517
Harmonic 5	Minimum	V3	Standard deviation	0.00513

Table 6.6: The 25 most important features in Dataset 3, ranked by a Random Forest in Experiment 3. The importance of each feature is the information gained by splitting the feature in a Decision Tree.

score.

The *Time step feature* column specifies whether it is the maximum value or minimum value within each time step interval that was extracted from the database. The *Aggregation method* column describes which aggregation method that was used to reduce the time series into a single value for that particular column. For example, the most important feature is the 10^{th} harmonic frequency in phase 2, where the minimum of the harmonic frequency has been extracted from the database for each interval in the time series. The whole sequence of minimum values has been aggregated into a single value by using the maximum operator on the sequence.

The top 25 features are all minimum thresholds of a harmonic frequency, aggregated by either calculating the standard deviation of the sequence, or by taking the maximum value, except for a single mean value. In fact, from Appendix B we can observe that the 115 most important features are all different aggregations of the minimum harmonic value per time step. Harmonic number 10 and 4 on different phases and lines make up 13 of the first 25 features. The line and phase voltages have no clear predominance in the features.

6.4.4 Experiment 4 Results

The model that achieved the highest MCC score on the validation set within each of the categories SVM, Random Forest, Feed-Forward Neural Network and Recurrent Neural Network were trained on the full training set and benchmarked on the test set. The results from the benchmarks are presented in table 6.7. The complete list of model performance metrics computed during grid search is included in Appendix D.

The Random Forest model achieved the highest accuracy and MCC out of all the machine learning models tested. The SVM achieved the lowest scores.

Figure 6.3 contains ROC curves for the tested models. A steep slope near the origin is preferable. The Random Forest has the steepest slope and the highest Area Under The Curve (AUC) out of the models tested.

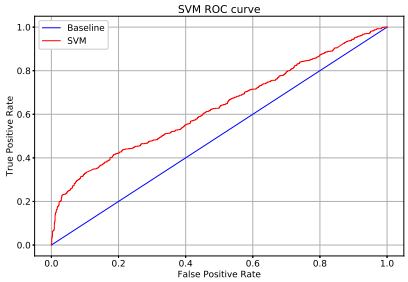
6.4.5 Experiment 5 Results

The Random Forest from Experiment 4 was trained on the full training set, before predicting on the test set. The samples in the test set were then divided into their fault category groups, and the accuracy within each category is listed in Table 6.8.

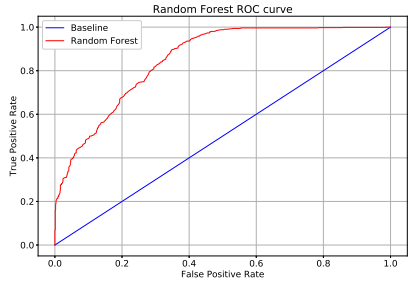
The model had the highest accuracy within the power interruption category, and performed the worst on voltage sags and RVC. The model labeled a non-fault as a fault in 19.8% of the samples.

Model	Parameters	Accuracy	MCC	AUC
SVM	Kernel = Radial basis function Penalty multiplier = 10	0.6077	0.2770	0.6311
Random Forest	Max depth = 20 Number of trees = 400 Split criterion = Information Gain	0.7423	0.4828	0.8522
Feed-forward Neural Network	First layer size = 128 Second layer size = 64 Third layer size = 64 Optimizer = Adam Loss function = Binary cross-entropy Batch size = 128 Number of epochs = 300	0.7083	0.4166	0.7551
Recurrent Neural Network	First bidirectional LSTM state size = 128 Second bidirectional LSTM state size = 64 Attention layer size = 64 Fully-connected layer size = 64 Optimizer = Adam Loss function = Binary cross-entropy Batch size = 128 Number of epochs = 300	0.6839	0.3661	0.7383

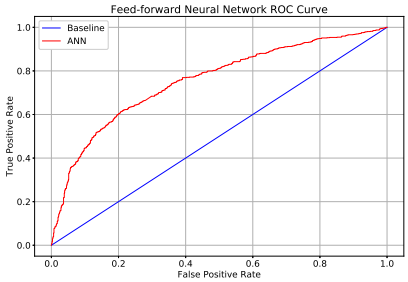
Table 6.7: Accuracy, Matthews Correlation Coefficient (MCC) and Area Under The Curve (AUC) scores for models tested in Experiment 4.



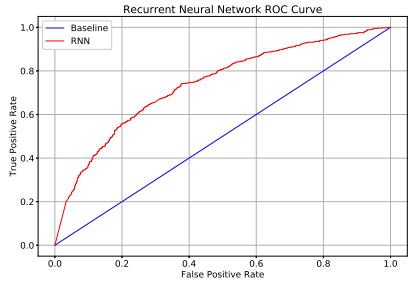
(a) SVM ROC curve in Experiment 4.



(b) Random Forest ROC curve in Experiment 4.



(c) Feed-forward Neural Network ROC curve in Experiment 4.



(d) Recurrent Neural Network ROC curve in Experiment 4.

Figure 6.3: Receiver Operating Characteristic (ROC) curves for the machine learning models tested in Experiment 4.

Sample category	**Accuracy**
Voltage sag	0.632
Ground faults	0.700
Rapid voltage changes	0.571
Power interruptions	0.865
Non-faults	0.802

Table 6.8: The accuracy of the Random Forest model from Experiment 4 within each fault category in Experiment 5.

6.4.6 Experiment 6 Results

The model with the highest MCC from Experiment 4 was trained on four different datasets, each consisting of samples from a single fault category. The performance on each dataset

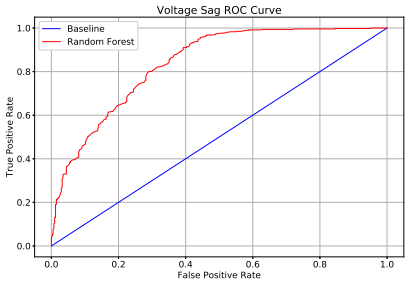
is presented in Table 6.9. The ROC curves within each fault category is displayed in Figure 6.4.

Fault type	**Accuracy**	**MCC**	**AUC**
Voltage sag	0.7133	0.4358	0.8381
Ground faults	0.7295	0.4652	0.8351
Rapid voltage changes	0.6603	0.3401	0.6980
Power interruptions	0.7613	0.5294	0.8317

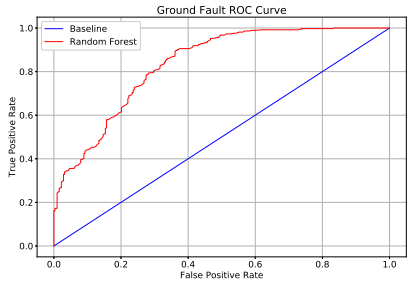
Table 6.9: Accuracy, Matthews Correlation Coefficient (MCC) and Area Under The Curve (AUC) scores achieved when training and predicting on single classes of faults in Experiment 6.

The Random Forest performed slightly better when predicting only power interruptions compared to the model predicting on all fault categories. Slightly worse performance was seen on voltage sags, with a larger decrease in performance in the rapid voltage change category. The model had close to the same performance on ground faults as when predicting on all fault categories.

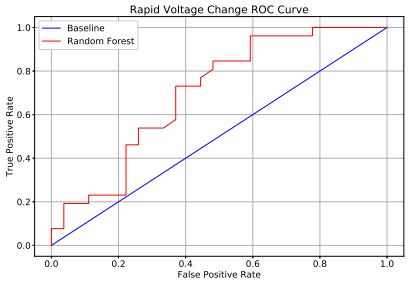
The ROC curves for voltage sags and ground faults are quite similar to the ROC curves the Random Forest produced on the whole dataset in Experiment 4. The curves for rapid voltage changes and power interruptions have a flatter curve near the origin.



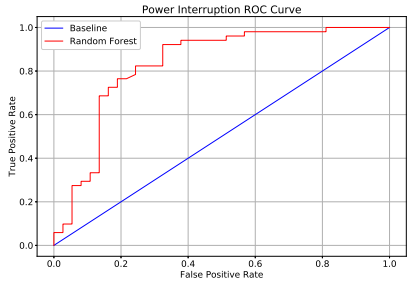
(a) ROC curve for a Random Forest trained and tested on voltage sag faults.



(b) ROC curve for a Random Forest trained and tested on ground faults.



(c) ROC curve for a Random Forest trained and tested on rapid voltage change faults.



(d) ROC curve for a Random Forest trained and tested on power interruption faults.

Figure 6.4: Receiver Operating Characteristic (ROC) curves for a Random Forest trained and tested on datasets consisting of faults from single fault categories in Experiment 6.

Chapter 7

Analysis

In this chapter we start out by discussing the results from the experiments described in Chapter 6. Section 7.1 covers the specific findings in each experiment, followed by Section 7.2 which puts the analysis into the broader context of this thesis.

7.1 Discussion

7.1.1 Experiment 1 - Feasibility test

Experiment 1 was performed to provide statistical proof that prediction of faults is possible with the voltage data available.

Under the assumption that a machine learning model has a 50% chance of classifying a random sample into either category in a balanced dataset with two outcome classes, we can use a binomial test to calculate the statistical significance of our experiment. Our null hypothesis H_0 is that our machine learning model has probability $P = \frac{1}{2}$ of classifying a sample into the correct category. Our alternative hypothesis H_a is that the model has probability $P > \frac{1}{2}$ of classifying correctly.

The model with the lowest accuracy in Experiment 1 was the SVM, which achieved an accuracy of 57.46%. Dataset 1 contained 8520 samples, which are independent and identically distributed random variables. 20% of the dataset was used as a test set, resulting in a total of 1704 samples for our binomial test. Out of these 1704 samples, 979 were correctly predicted.

By using the binomial distribution $B(N = 1704, P = \frac{1}{2})$, we calculate the probability of getting 979 or more correctly predicted samples in an experiment where the true probability of predicting the correct class is $\frac{1}{2}$. The probability of this occurring is $4.1 \cdot 10^{-10}$.

By using the statistical significance level of 5%, we can reject the null hypothesis and accept H_a. Our machine learning models can correctly classify a sample with probability $P > \frac{1}{2}$.

7.1.2 Experiment 2 - Harmonic frequency presence

Experiment 2 was performed to provide insight into which harmonic frequencies that are present in fault and non-fault signals.

From Figure 6.2 it is clear that there are 5 harmonic frequencies that are more prevalent in the voltage signal than other frequencies. Harmonic number 3, 5, 7 and 11 are all present in more than 40% of all time steps, while Harmonic number 13 is present in about 15% of the signal, for both faults and non-faults. These are all odd harmonic frequencies, which are more common in three-phase systems, as even harmonic values are usually canceled out by the three-phase system. Around harmonic number 33 and 67, there is a sudden increase in harmonic presence.

There is no clear difference in which harmonics appear in signals leading up to a fault or in non-faulty signals. The distribution is quite similar between the two, where some frequencies are a few percentages more common in faults, and other frequencies are more common in non-faults. We hypothesise that the mere presence of harmonic frequencies is not an indicator of an upcoming fault, but rather that the amplitude and patterns in the harmonic frequencies over time are the main indicators.

Columns that were present in more than 1% of the dataset were included in Dataset 3, to be used for training machine learning models. The cut-off threshold was chosen after the harmonic counts were available, in order to filter out most of the frequencies which only appeared in a few samples. The low frequency counts can be regarded as noise, and may lead to overfitting in models by remembering samples with these specific harmonic frequencies. Setting the cut-off threshold at a somewhat higher percentage might have removed more noise, but would quickly reduce the number of features to only five harmonics, loosing the information from the columns with a somewhat higher count around harmonic 33 and 67.

As stated in Table 5.2, harmonic frequencies up to and including the 512$^{\text{th}}$ harmonic is available in the database. Due to limitations in the software used to generate fault samples from the database, generating all 512 frequencies for more than 30 minutes with a sample rate of 1Hz is not feasible. Because of this, only the first 256 harmonics were included in the dataset. Based on the full table of counted values in Appendix A, we can see that no harmonic frequencies above the 97$^{\text{th}}$ frequency is present in the dataset. We thus hypothesise that this trend continues on until the 512$^{\text{th}}$ harmonic, but we acknowledge that there

is no guaranteed way to be sure without actually generating the harmonic frequencies and counting the number of non-zero values.

7.1.3 Experiment 3 - Feature importance

As we can see from Table 6.6 and Appendix B, the 112 most important features were different aggregations of the minimum harmonic value per time step. This is strongly indicating that inspecting the minimum value of the harmonic signals is key to predicting faults.

From Experiment 2 we already know that some of the harmonic values are present with non-zero amplitude in almost 80% of the time steps, so in practice there is a clear presence of harmonic distortion, even in non-faulty signals. Experiment 3 only looks at which features are the most important attributes to split to divide the dataset into its two categories, it does not say which side of the split the two categories fall into. Theory on harmonics in power systems, state that an ideal power system should have zero amplitude on all harmonic frequencies, except for the fundamental frequency. We thus assume that a lower amplitude in harmonic frequencies are generally a good indicator of a healthy signal, and higher amplitudes are indicators of a faulty signal. The splits in the Decision Trees are believed to reflect this behaviour.

The aggregation methods maximum value, mean value and standard deviation made up all of the 112 most important features, with 40, 36 and 36 features respectively. The first feature made with minimum value aggregation appeared as the 116^{th} most important feature.

Aggregating a high maximum value from the minimum values of each time step is an indicator that the signal has had a high degree of distortion for a whole sample interval period, in our case 1 second. Such a distortion can be regarded as a minor disturbance on the line, which eventually leads to a larger fault.

The mean and standard deviation of the minimum values can also be regarded as indicators of an upcoming fault, where a high mean value can indicate a lasting distortion in the signal, while the standard deviation measures changes in the harmonic values over time. Ideally, the power system should have a stable signal, and the standard deviation is a measure of how much the signal fluctuates.

The minimum value aggregation is believed to give little useful information when predicting faults. The interval used for aggregation was a window of 10 minutes, with a sample rate of 1 sample per second. A window of 1 second with zero amplitude in a harmonic frequency would result in the overall aggregation returning a zero amplitude. Unless the signal is severely distorted during the whole period, the probability of such an interval appearing is high, effectively removing all other information from the aggregation.

7.1.4 Experiment 4 - Model performance test

Random Forests achieved the best scores in accuracy, MCC and AUC. The ROC curve produced by the Random Forest also has the steepest rise around the origin out of the four models tested. We can thus conclude that the Random Forest is the best machine learning model for fault prediction out of the four models that were tested as part of this thesis.

The steep rise of the ROC curve from the Random Forest is beneficial when it comes to practical applications for fault prediction. A system which provides too many false alarms will result in lack of trust in the system by the grid operators. By setting the threshold for an alarm close to 100% confidence, we can see from Figure 6.3 that the Random Forest is able to predict almost 20% of the fault events with a minimal amount of false negatives.

The threshold for prediction would need to be tweaked in an actual application, in order to determine a tolerable ratio of true positives to false positives. Another solution could be to set two thresholds on the output, where the highest threshold was used to directly trigger an alarm, while the second, lower threshold is used to flag a voltage signal for further processing with other machine learning models or automated analysis tools.

Feed-forward Neural Networks, Recurrent Neural Networks and Random Forests have in common that they are universal function approximators. They are in theory able to approximate all possible functions, with arbitrary precision. This does not necessarily imply that neural networks provide the best solution to any problem that can be modelled as a continuous function. To be able to approximate a complicated function, the number of nodes in the hidden layers of the networks increases, which require more training samples in order to not overfit to the training data, and to still be able to approximate the function well.

A Random Forest can approximate any function by creating Decision Trees that have no limit on tree depth, and keep splitting until all samples in a sub-tree are in a group of samples with a single class. It is still limited by the trends and data represented in the training samples, but it contains fewer parameters that needs to be tuned with increasing amounts of data. The Feed-forward and Recurrent Neural Network consisted of 33 857 and 161 415 trainable parameters respectively, and with a dataset of 6924 samples for training, the task of approximating the prediction function is similar to fitting a 10^{th} degree polynomial to a single point; there is simply not enough data to fit the function approximation to.

An interesting find is that the preprocessing method which completely removed the spatial dimension from the signal performed better in combination with the Random Forest and the Feed-forward Network than the partially spatial-preserving preprocessing in combination with the Recurrent Network. This might be an indicator that a gradual build-up of a faulty signal is not necessarily needed in order to predict faults, and that it is sufficient that a fault signature is present somewhere in the signal, regardless of the spatial sequence of events. However, because of the ratio between parameters and training samples in the Recurrent Network, there might be a lack of data that led to the lower performance, not that the sequence of fault indicators are indifferent to the prediction process.

Out of the four models tested, the SVM achieved the lowest scores in all recorded metrics. The ROC curve shows that the ratio between true positives and false positives are close to the baseline, with only a slight trend towards more true positives than false positives. The model performed better than the SVM used in Experiment 1, but is still beaten by all other models tested. The number of support vectors in the SVM ended up as 5997 vectors. This indicates that the SVM is not able to split the feature space into separate regions for the respective classes, but that most of the training samples lie on the border of the decision line.

7.1.5 Experiment 5 - Fault category performance

From Table 6.8 we can observe that there is a clear difference in accuracy between the fault categories. Power interruptions achieved the highest accuracy of 86.5%, while RVCs achieved an accuracy of 57.1%. Ground faults achieved an accuracy close to the mean accuracy across the different categories, while voltage sags performed worse than average, with an accuracy of 63.2%.

The results indicates that some faults are easier to detect than others. No weighting of different fault categories was performed when training the Random Forest model, so there is no preferential bias in the model, other than trying to achieve the highest possible accuracy across all categories. Power interruption faults are clearly easier to detect than other faults, with properties in the signal leading up to the fault that are able to be detected by the Random Forest in most cases.

It must be noted that the category that performed the best and the category that performed the worst had the lowest amount of labeled faults within their categories. Only 220 power interruptions and 132 RVCs are part of the whole dataset, compared to 2178 and 1710 samples for voltage sags and ground faults respectively. Out of these faults, only 20% were included in the test set, on which the metrics in Experiment 5 are based on. The randomly chosen samples that constitutes the test set can thus have influenced the score in either a positive or negative way, by not being representative for the distribution of samples in the full fault category.

Another observation from the results is the number of false positive faults detected. only 80.2% of non-faults were classified as non-faults, while 19.8% of the were incorrectly classified as faults. As discussed in Section 7.1.4, a false positive rate close to 20% is too high for any system that aims to help grid operators in practical decision making. However, the threshold used in this experiment was set to a majority vote among the trees in the Random Forest, and a higher threshold for fault detection will reduce the number of false positives, as observed in the results from Experiment 4.

7.1.6 Experiment 6 - Fault category specialization

Some of the models trained on individual categories of faults achieved results close to the performance of the model trained on the whole dataset. Voltage sags and ground faults achieved an accuracy a few percentage points below the previous results, while power interruptions performed better than the initial model. The model trained on RVCs performed he worst out of the four models with 66% accuracy, more than 8 percentage points below the initial model.

Combining the results from Experiment 5 with the results from Experiment 6, we can see that accuracy on the RVC category was the lowest both when trained on a full dataset and when trained on only RVC faults. We hypothesis that there are no indicators or fewer indicators in the voltage signal than in other categories leading up to these type of faults.

When looking at the model trained on only power interruption faults, we can see that the accuracy and MCC was higher than the overall accuracy and MCC of the model trained on all fault categories. However, comparing the results to the actual performance within the power interruption category from Experiment 5, there is actually a loss in performance within the category. As mentioned in Section 7.1.5, there are fewest samples within the RVC and power interruption categories, which might influence the test set by not being representative of the underlying distribution of samples.

From Figure 6.4 we can see that the ROC curves for voltage sags and ground faults are similar to the ROC curve for the Random Forest model in Figure 6.3. This shows that the models are able to predict some faults with a high degree of certainty, allowing for high thresholds to filter out false positives.

The curves for RVCs and power interruptions are significantly different to the previous Random Forest curve, were voltage sags have a lower Area Under the Curve, and the power interruption curve has few samples where the model is certain of a coming fault.

We argue that making a Random Forest model specifically for each category is feasible for voltage sags and ground faults, but more data is needed in order to determine whether the same applies for RVCs and power interruption faults. Further research is also needed in order to determine the behaviour by models trained on a single fault type and presented with a fault of another type.

7.1.7 Online prediction

The fact that the Random Forest model achieved the highest scores in all recorded metrics is positive for the further development of an application for fault prediction.

Training a Random Forest on a large dataset is parallelizable, and can be sped up by adding additional CPU cores to a machine. The forest used in Experiment 4 was trained

on an Xeon Gold 6126 CPU, with 24 logical cores, clocked at 2.4 GHz with a maximum frequency of 3.7 GHz utilizing Intel's Turbo Boost Technology.

By only utilizing 4 of the cores available, the model used a mean of 3213 milliseconds, with a standard deviation of 34 milliseconds, recorded over 7 runs to train on the training data. Predicting a single sample used a mean of 42 milliseconds, with a standard deviation of 2 milliseconds. Aggregating the features used by the Random Forest takes 17 milliseconds with a standard deviation of 2 milliseconds.

The short time span required to extract and predict a sample makes the application usable for online prediction of faults. With the warning horizon of 10 minutes used in this thesis, the rate of prediction can be set to every second, or as soon as new data is available, while still maintaining a warning horizon close to 10 minutes.

7.2 Contributions

The main contribution of this thesis is research into the field of power system fault prediction, specifically into feature engineering, machine learning models for prediction and performance analysis for different fault categories.

7.2.1 Research Questions

The following research questions have been answered

Can machine learning methods predict upcoming faults on power lines by analyzing voltage measurement data?

Experiment 1 confirms that it is possible to predict upcoming faults with a statistically significant result. The model with the lowest accuracy out of the three models tested proved to be better than random guess. Further models have been tested in Experiment 4, all which provided a higher accuracy than the initial model with the lowest score.

What attributes in a voltage signal are suited for predicting faults on power lines?

Experiment 2 and 3 provided insight into which attributes in a voltage signal that are useful for predicting power system faults, and which harmonic frequencies that are present in both faulty and non-faulty signals. Monitoring the minimum values of the harmonic frequencies over time combined with multiple aggregations methods provided good results in combination with Random Forests and Neural Networks.

What is the performance of some machine learning algorithms at predicting faults on power lines?

In Experiment 4, SVMs, Random Forests, Feed-forward Neural Networks and Recurrent Neural Networks were tested on voltage prediction with a 10 minute prediction horizon. SVMs achieved the lowest accuracy and MCC scores, while Random Forests achieved the highest scores, with an accuracy of 74.2% and an MCC of 0.48.

Are some types of faults easier to predict than others?

Experiment 5 shows that some type of faults are easier to predict than others. Power interruptions were correctly predicted with an accuracy of 86.5%, while RVCs were correctly predicted in only 57.1% of the cases. By training models on one single category of faults, the prediction accuracy within each category rose, except for the model trained on power interruption faults.

7.2.2 Dynamic Dataset Generator

In addition to performing experiments to answer the research questions in this thesis, software for generating the dataset used in this thesis was developed.

In order to automate the data extraction process, a Command Line Interface (CLI) was developed to interface with the Dynamic Data Generator (DDG) software developed by SINTEF Energy, enabling queuing of data extraction jobs. The CLI allows for automatic job creation, without the need of a graphical user interface such as DDG GUI, which enables DDG to be fully incorporated into an automated pipeline in the future.

The backend engine for data generation, DDG Backend, was rewritten to support the new CLI. The backend is now more error resistant, with more detailed logging of error events and feedback to the user if an error occurs while generating datasets. The backend now retries multiple times to generate a time series sample if it fails on the first try. This results in more robust generation of datasets, with fewer samples missing from the final datasets.

Chapter 8

Conclusion

With an ever-increasing demand for power and electricity in the modern world, providing a stable source of power through the power grid is of utmost importance. Aging infrastructure and the introduction of new sources of energy leads to increased strain, resulting in costs averaging more than 800 million NOK per year for maintenance and repairs in the Norwegian power grid.

Prediction of faults using high resolution voltage measurement data from the Norwegian power grid has been performed. The initial results show that machine learning models such as Support Vector Machines, Random Forests and Neural Networks are capable of predicting faults with a 10 minute prediction horizon.

We have shown that feature engineering using the minimum thresholds of harmonic frequencies in the voltage signal increases the performance of the machine learning models compared to using the mean values of harmonics over time. By aggregating the signal along the spatial dimension using operators such as standard deviation, mean value and maximum value, the models are able to achieve a higher accuracy and Matthews Correlation Coefficient.

Multiple variations of Support Vector Machines, Random Forests and Neural Networks have been tested in combination with the improved feature engineering methods. The Random Forest achieved the highest scores among the models, with an accuracy of 74.2% and a Matthews Correlation Coefficient of 0.48. The model shows great potential for use in applications for online fault prediction, requiring less than 70 milliseconds for feature engineering and prediction of a single sample.

The prediction accuracy within different categories of faults varies. The Random Forest model achieves the highest accuracy in the fault category power interruptions, and the

lowest accuracy on rapid voltage changes. Training a Random Forest on a single category of faults increases the accuracy within the category compared to the accuracy achieved when training on a dataset of multiple categories.

8.1 Future Work

During the course of this thesis, multiple ideas and problems that could be further addressed were discovered. Due to time constraints and that some of the problems are outside the scope of this thesis, the problems remain open for future work. This section lists some of the areas that can potentially be looked into by EarlyWarn or other researchers working on similar problems as the ones addressed in this thesis.

8.1.1 Prediction horizon

The prediction horizon chosen for this thesis was set to 10 minutes. The value was chosen based on early experiments and research by EarlyWarn, where also longer prediction horizons seemed plausible. Using the same algorithms as presented in this thesis, a longer and shorter prediction horizon should be tested and compared to the performance of the initial 10 minute threshold.

A longer time interval for aggregations could also be tested, in order to detect faults where the early indicators are far apart in the signal leading up to the event. For longer periods of time, other aggregation methods or models that support time time series data, such as Recurrent Neural Networks, should be further explored.

8.1.2 Alarm outputs

The machine learning models tested in this thesis output the probability of an upcoming fault. Setting a threshold close to 100% confidence might lead to few false positive alarms, but at the same time lead to a large number of faults not being detected. The system created by The University of Sao Paulo and Enerq, as explained in Section 4.4, used a two-stage alarm system, where the tier 1 alarm could be escalated into a tier 2 alarm after further processing of the signal. A similar system could be tested out by EarlyWarn, but instead of the first alarm notifying the grid operators directly, the signal would instead be flagged for further processing, potential triggering an alarm.

Guo and Milanović used a security index of alertness instead of a boolean output, as explained in Section 4.3. This idea could be applied to the outputs of a prediction system, where multiple threshold at different confidence levels made out the alertness levels of the output signal. This way, the grid operators could monitor the overall predicted state of the

system, and not be taken by surprise when the system suddenly outputs an prediction of an upcoming fault.

8.1.3 Live system test

The models in this thesis has not been tested as part of an application for prediction on a live system. The next natural step is to create a prototype where the output of the model is used as part of a system for live prediction at a grid operations center, possibly without direct feedback to the grid operators in the first iterations of the system.

In order to prepare for this, the model should be tested on a time series with labeled faults which are not part of the training set used to train the model, to measure the performance when used as an online system. Setting the alarm threshold to a high degree of confidence will be crucial, as even a low ratio of false positives will lead to a high number of alarms when used to predict on power lines over long intervals, with multiple outputs per minute.

Bibliography

[1] Christian Andre Andresen. Earlywarn. https://www.sintef.no/en/projects/earlywarn/. Accessed: 2018-09-26.

[2] Norges vassdrags-og energidirektorat. Overview of Norway's electricity history. http://publikasjoner.nve.no/rapport/2017/rapport2017_15.pdf, 2017. Accessed: 2018-09-27.

[3] Nur IqtiyaniIlham, M Hasanuzzaman, and M Hosenuzzaman. European smart grid prospects, policies, and challenges. *Renewable and Sustainable Energy Reviews*, 67:776–790, 2017.

[4] Olje og energidepartementet. Fakta energi- og vannressurser i Norge. https://www.regjeringen.no/contentassets/fd89d9e2c39a4ac2b9c9a95bf156089a/1108774830_897155_fakta_energi-vannressurser_2015_nett.pdf. Accessed: 2018-12-05.

[5] Norges vassdrags-og energidirektorat. Avbrotsstatistikk 2017. http://publikasjoner.nve.no/rapport/2018/rapport2018_64.pdf, 2018. Accessed: 2018-10-26.

[6] International Energy Agency. *World Energy Outlook*, chapter 1, pages 38–43. IEA Publications, 2018.

[7] Statnett SF. Strømnettets historie. http://www.statnett.no/nettutvikling/nettplan-stor-oslo/om-nettplanen/stromnettets-historie/. Accessed: 2018-10-26.

[8] Statistisk sentralbyrå. Kraftinvesteringer i støtet. https://www.ssb.no/energi-og-industri/artikler-og-publikasjoner/kraftinvesteringer-i-stotet/. Accessed: 2018-10-26.

[9] Statistisk sentralbyrå. Investeringer i olje og gass, industri, bergverk og kraftforsyning. https://www.ssb.no/energi-og-industri/statistikker/kis. Accessed: 2018-10-26.

[10] Statnett. Årsstatistikk 2017. http://www.statnett.no/global/dokumenter/kraftsystemet/systemansvar/feilstatistikk/%C3%85rsstatistikk%202017%2033_420%20kv.pdf. Accessed: 2018-10-26.

[11] Gerd Kjølle. KILE. https://www.sintef.no/kile-kvalitetsjusterte-inntektsrammer-ved-ikke-lev/. Accessed: 2018-06-08.

[12] Vegard Hellem and Vemund Mehl Santi. A study of fault analysis in power grids using machine learning. https://github.com/santi/tdt4900-master-thesis/blob/master/project-report/a-study-of-fault-analysis-in-power-grids-using-machine-learning.pdf. Accessed: 2019-05-19.

[13] Paul Horowitz and Winfield Hill. *The Art of Electronics*, pages 1–2, 14, 1099. Cambridge University Press, New York, NY, USA, 3 edition, 2015.

[14] MJ Matabaro and G Atkinson-Hope. Efficiency of an alternating current transmission line converted into a direct current system. In *Industrial and Commercial Use of Energy Conference (ICUE), 2013 Proceedings of the 10th*, pages 1–6. IEEE, 2013.

[15] International Electrotechnical Commision. World plugs. https://www.iec.ch/worldplugs/list_bylocation.htm. Accessed: 2019-04-30.

[16] Vijay Venu Vadlamudi. Fundamentals of Power Systems Refresher for TET4115 (Power System Analysis), 2018.

[17] Sergey Makarov, Reinhold Ludwig, and Stephen Bitar. *Practical Electrical Engineering*. Springer International Publishing, 2016.

[18] Mihaela Albu and Gerald T. Heydt. On the use of RMS values in power quality assessment. *IEEE Transactions on Power Delivery*, 18(4):1586–1587, 2003.

[19] Richard Dorf and James Svoboda. *Introduction to Electric Circuits*, pages 502, 871. John Wiley and Sons, Inc, 8 edition, 2010.

[20] Henrik Kirkeby. Automatisk hendelsesanalyse. https://www.sintef.no/globalassets/project/nef-tm-2017/rapporter-2017/sesjon-1-5---22---automatisk-hendelsesanalyse-a-ha_nyversjon.pdf, 2017. Accessed: 2019-05-14.

[21] J. Barros and E. Pérez. Limitations in the Use of R.M.S. Value in Power Quality Analysis. In *IEEE Instrumentation and Measurement Technology Conference Proceedings*, pages 2261–2264, 2006.

[22] E.M. Stein and R. Shakarchi. *Fourier Analysis: An Introduction*. Princeton University Press, 2003.

[23] Lucas Illing. Fourier analysis. https://www.reed.edu/physics/courses/physics331.f08/pdf/fourier.pdf, 2008. Accessed: 2018-10-14.

[24] L. Salasnich. *Quantum Physics of Light and Matter: A Modern Introduction to Photons, Atoms and Many-Body Systems*, pages 229 – 232. UNITEXT for Physics. Springer International Publishing, 2014.

[25] Julius O. Smith. *Mathematics of the Discrete Fourier Transform (DFT)*, chapter 1. W3K Publishing, 2007.

[26] Juhan Nam. Super-resolution spectrogram using coupled PLCA. https://ccrma.stanford.edu/~juhan/super_spec.html, 2010. Accessed: 2018-12-06.

[27] fred harris. On the use of windows for harmonic analysis with the discrete fourier transform. *Proceedings of the IEEE*, 66:51 – 83, 02 1978.

[28] Tony R. Kuphaldt. Lessons In Electric Circuits, Volume II AC. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.690.8067, 2007. Accessed: 2019-05-14.

[29] D. Shmilovitz. On the definition of total harmonic distortion and its effect on measurement interpretation. *IEEE Transactions on Power Delivery*, 20(1):526–528, Jan 2005.

[30] Hai-Man Chung and Don C. Lawton. Amplitude responses of thin beds: Sinusoidal approximation versus Ricker approximation. *GEOPHYSICS*, 60(1):223–230, 1995.

[31] Yu-Hsiang Wang. The Tutorial: S Transform, 2010.

[32] Edith Clarke. Simultaneous faults on three-phase systems. *Transactions of the American Institute of Electrical Engineers*, 50(3):919–939, 1931.

[33] Yong Sheng and Steven M. Rovnyak. Decision tree-based methodology for high impedance fault detection. *IEEE Transactions on Power Delivery*, 19(2):533–536, 2004.

[34] M. Sedighizadeh, A. Rezazadeh, and Nagy I. Elkalashy. Approaches in high impedance fault detection a chronological review. *Advances in Electrical and Computer Engineering*, 10(3):114–128, 2010.

[35] Amin Ghaderi, Herbert L. Ginn III, and Hossein Ali Mohammadpour. High impedance fault detection: a review. *Electric Power Systems Research*, 143:376–388, 2017.

[36] Joseph Seymour and Terry Horsley. The seven types of power problems. *APC, USA*, 2005.

[37] Thomas M Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997.

[38] Yann LeCun, Yoshua Bengio, and Geoffrey E. Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[39] Leo Breiman. Bagging predictors. *Machine Learning*, 24(2):123–140, 1996.

[40] Robert E. Schapire. The strength of weak learnability. *Machine Learning*, 5(2):197–227, 1990.

[41] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. http://www.deeplearningbook.org.

[42] Favio Vázquez. Deep learning made easy with deep cognition. https://becominghuman.ai/deep-learning-made-easy-with-deep-cognition-403fbe445351. Accessed: 2018-12-07.

[43] Antonio Salmerón, Rafael Rumí, Helge Langseth, Thomas D Nielsen, and Anders L Madsen. A review of inference algorithms for hybrid Bayesian networks. *Journal of Artificial Intelligence Research*, 62:799–828, 2018.

[44] Padhraic Smyth. Hidden Markov models for fault detection in dynamic systems. *Pattern recognition*, 27(1):149–164, 1994.

[45] Sreerama K Murthy. Automatic construction of decision trees from data: A multi-disciplinary survey. *Data mining and knowledge discovery*, 2(4):345–389, 1998.

[46] Thomas G. Dietterich. Ensemble methods in machine learning. In *Proceedings of the First International Workshop on Multiple Classifier Systems*, MCS '00, pages 1–15, London, UK, UK, 2000. Springer-Verlag.

[47] Andy Liaw, Matthew Wiener, et al. Classification and regression by random forest. *R news*, 2(3):18–22, 2002.

[48] Nello Cristianini and John Shawe-Taylor. *An Introduction to Support Vector Machines: And Other Kernel-based Learning Methods*. Cambridge University Press, New York, NY, USA, 2000.

[49] Andreas Marouchos. DS Fundamentals: Support Vector Machines. https://dataincognita.net/2017/12/06/ds-fundamentals-p2-support-vector-machines/. Accessed: 2018-10-12.

[50] K. . Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf. An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, 12(2):181–201, March 2001.

[51] Duane Q. Nykamp. Dynamical system definition. https://mathinsight.org/definition/dynamical_system. Accessed: 2018-09-27.

[52] Elizabeth D Liddy. Natural language processing. In *Encyclopedia of Library and Information Science*, volume 18. Marcel Decker, 2 edition, 2001.

[53] Niklas Donges. Recurrent Neural Networks and LSTM. https://towardsdatascience.com/recurrent-neural-networks-and-lstm-4b601dd822a5. Accessed: 2018-09-28.

[54] Dong Chan Lee. Automatic power quality monitoring with recurrent neural network. Master's thesis, University of Toronto, 2016.

[55] D. M. W. Powers. Evaluation: From precision, recall and f-measure to roc., informedness, markedness & correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011.

[56] Sabri Boughorbel, Fethi Jarray, and Mohammed El-Anbari. Optimal classifier for imbalanced data using Matthews Correlation Coefficient metric. *PLOS ONE*, 12(6):1–17, 06 2017.

[57] Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1):19–67, Feb 2005.

[58] M H Zweig and G Campbell. Receiver-operating characteristic (roc) plots: a fundamental evaluation tool in clinical medicine. *Clinical Chemistry*, 39(4):561–577, 1993.

[59] Christian Andre Andresen, Bendik Torsæter, Hallvar Haugdal, and Kjetil Uhlen. Fault detection and prediction in smart grids. *IEEE International Workshop on Applied Measurements for Power Systems*, 2018.

[60] Ana Claudia Barros, Mauro S Tonelli-Neto, José Guilherme Magalini Santos Decanini, and Carlos Roberto Minussi. Detection and classification of voltage disturbances in electrical power systems using a modified euclidean artmap neural network with continuous training. *Electric Power Components and Systems*, 43(19):2178–2188, 2015.

[61] Om Prakash Mahela, Abdul Gafoor Shaik, and Neeraj Gupta. A critical review of detection and classification of power quality events. *Renewable and Sustainable Energy Reviews*, 41:495–505, 2015.

[62] Atish K Ghosh and David L Lubkeman. The classification of power system disturbance waveforms using a neural network approach. *IEEE Transactions on Power Delivery*, 10(1):109–115, 1995.

[63] AM Gaouda, MMA Salama, MR Sultan, and AY Chikhani. Power quality detection and classification using wavelet-multiresolution signal decomposition. *IEEE Transactions on power delivery*, 14(4):1469–1476, 1999.

[64] D Granados-Lieberman, Rj Romero-Troncoso, Ra Osornio-Rios, A Garcia-Perez, and E Cabal-Yepez. Techniques and methodologies for power quality analysis and disturbances classification in power systems: a review. *IET Generation, Transmission & Distribution*, 5(4):519–529, 2011.

[65] H Siahkali. Power quality indexes for continue and discrete disturbances in a distribution area. In *Power and Energy Conference, 2008. PECon 2008. IEEE 2nd International*, pages 678–683. IEEE, 2008.

[66] Florence Choong, MBI Reaz, and F Mohd-Yasin. Advances in signal processing and artificial intelligence technologies in the classification of power quality events: a survey. *Electric Power Components and Systems*, 33(12):1333–1349, 2005.

[67] Przemysław Janik and Tadeusz Lobos. Automated classification of power-quality disturbances using SVM and RBF networks. *IEEE Transactions on Power Delivery*, 21(3):1663–1669, 2006.

[68] Peter GV Axelberg, Irene Yu-Hua Gu, and Math HJ Bollen. Support vector machine for classification of voltage disturbances. *IEEE Transactions on power delivery*, 22(3):1297–1303, 2007.

[69] Shunfan He, Kaicheng Li, and Ming Zhang. A real-time power quality disturbances classification using hybrid method based on S-transform and dynamics. *IEEE transactions on instrumentation and measurement*, 62(9):2465–2475, 2013.

[70] Raj Kumar, Bhim Singh, DT Shahani, Ambrish Chandra, and Kamal Al-Haddad. Recognition of power-quality disturbances using S-transform-based ANN classifier and rule-based decision tree. *IEEE Transactions on Industry Applications*, 51(2):1249–1258, 2015.

[71] Prakash K Ray, Soumya R Mohanty, Nand Kishor, and João PS Catalão. Optimal feature and decision tree-based classification of power quality disturbances in distributed generation systems. *IEEE Transactions on Sustainable Energy*, 5(1):200–208, 2014.

[72] Ebrahim Balouji, Irene YH Gu, Math HJ Bollen, Azam Bagheri, and Mahmood Nazari. A LSTM-based deep learning method with application to voltage dip classification. In *Harmonics and Quality of Power (ICHQP), 2018 18th International Conference on*, pages 1–5. IEEE, 2018.

[73] Ruisheng Diao, Kai Sun, Vijay Vittal, Robert J O'Keefe, Michael R Richardson, Navin Bhatt, Dwayne Stradford, and Sanjoy K Sarawgi. Decision tree-based online voltage security assessment using PMU measurements. *IEEE Transactions on Power Systems*, 24(2):832–839, 2009.

[74] M. Negnevitsky, N. Tomin, V. Kurbatsky, D. Panasetsky, A. Zhukov, and C. Rehtanz. A random forest-based approach for voltage security monitoring in a power system. In *2015 IEEE Eindhoven PowerTech*, pages 1–6, June 2015.

[75] Aleksey Zhukov, Nikita Tomin, Denis Sidorov, Daniil Panasetsky, and Vadim Spirayev. A hybrid artificial neural network for voltage security evaluation in a power system. In *Energy (IYCE), 2015 5th International Youth Conference on*, pages 1–8. IEEE, 2015.

[76] Amr M Ibrahim and Noha H El-Amary. Particle swarm optimization trained recurrent neural network for voltage instability prediction. *Journal of Electrical Systems and Information Technology*, 5(2):216–228, 2018.

[77] T. Guo and J. V. Milanović. On-line prediction of transient stability using decision tree method — sensitivity of accuracy of prediction to different uncertainties. In *2013 IEEE Grenoble Conference*, pages 1–6, June 2013.

[78] Mohammed R Qader, Math HJ Bollen, and Ron N Allan. Stochastic prediction of voltage sags in a large transmission system. *IEEE Transactions on Industry Applications*, 35(1):152–162, 1999.

[79] N. Kagan, N. M. Matsuo, E. L. Ferrari, V. Vega-García, J. C. García-Arias, T. P. Souza, A. B. Neto, L. B. Chaves, F. S. Ide, L. A. F. Carvalho, and A. K. Suematsu. Computerized system for detection of high impedance faults in MV overhead distribution lines. In *17th International Conference on Harmonics and Quality of Power (ICHQP)*, pages 992–997, Oct 2016.

[80] Alicia Valero Masa. High impedance fault detection method in multi-grounded distribution networks. Master's thesis, École polytechnique de Bruxelles - Université, 2012.

[81] A. M. El-Hadidy and C. Rehtanz. Early detection of tree faults. In *22nd International Conference and Exhibition on Electricity Distribution (CIRED 2013)*, pages 1–5, June 2013.

[82] T. Weckesser, H. Jóhannsson, and T. Van Cutsem. Early prediction of transient voltage sags caused by rotor swings. In *2014 IEEE PES General Meeting — Conference Exposition*, pages 1–5. IEEE, July 2014.

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

[84] B. D. Russell and C. L. Benner. Intelligent systems for improved reliability and failure diagnosis in distribution systems. *IEEE Transactions on Smart Grid*, 1(1):48–56, June 2010.

[85] K. Manivinnan, C. L. Benner, B. D. Russell, and J. A. Wischkaemper. Automatic identification, clustering and reporting of recurrent faults in electric distribution feeders. In *2017 19th International Conference on Intelligent System Application to Power Systems (ISAP)*, pages 1–6, Sep. 2017.

[86] F. Xiao and Q. Ai. Data-Driven Multi-Hidden Markov Model-Based Power Quality Disturbance Prediction that Incorporates Weather Conditions. *IEEE Transactions on Power Systems*, 34(1):402–412, Jan 2018.

[87] Elspec Ltd. Power Quality Analyzer. https://cuthbertsonlaird.co.uk/power-quality/328-blackbox-g4500.html. Accessed: 2018-10-04.

[88] Helge Seljeseth. Spenningskvalitetsmålinger – nyttig ved feilanalyse? https://www.energinorge.no/kurs-og-konferanser/2016/q4/fasit-dagene/dag-1/. Accessed: 2019-05-20.

[89] Forskrift om systemansvaret i kraftsystemet. https://lovdata.no/dokument/SF/forskrift/2002-05-07-448. Accessed: 2018-11-06.

[90] Statnett SF. Brukerveiledning for FASIT-rapportering i Fosweb. https://www.statnett.no/globalassets/for-aktorer-i-kraftsystemet/systemansvaret/fosweb/fosweb-fasit-rapportering---brukerveiledning.pdf. Accessed: 2018-06-04.

[91] Elspec LTD. G4400 - 3-phase class a power quality analyzer. https://www.elspec-ltd.com/metering-protection/power-quality-analyzers/g4400-power-quality-analyzer/. Accessed: 2019-05-20.

[92] Matthias Hofmann, Helge Seljeseth, Gro Holst Volden, and GH Kjølle. Study on estimation of costs due to electricity interruptions and voltage disturbances. *SINTEF Energy Rearch*, 2010.

[93] B. G. Hennessy and Boris Kulikov. *The boy who cried wolf*. New York: Simon & Schuster Books for Young Readers, New York, NY, USA, 2006.

[94] L. Xu, M. Chow, and L. S. Taylor. Data mining and analysis of tree-caused faults in power distribution systems. In *2006 IEEE PES Power Systems Conference and Exposition*, pages 1221–1227, Oct 2006.

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

[96] François Chollet et al. Keras. https://keras.io, 2015.

[97] Bruno Aquino. Kaggle - VSB Power Line Fault Detection. https://www.kaggle.com/braquino/5-fold-lstm-attention-fully-commented-0-694. Accessed: 2019-06-01.

Appendix A

Experiment 2 Results

Harmonic frequency	Percentage of non-zero values in faults	Percentage of non-zero values in non-faults
Harmonic 1	0.0%	0.0%
Harmonic 2	2.6%	2.0%
Harmonic 3	49.1%	46.7%
Harmonic 4	2.0%	1.9%
Harmonic 5	77.5%	74.3%
Harmonic 6	1.4%	1.3%
Harmonic 7	79.3%	77.9%
Harmonic 8	0.4%	0.3%
Harmonic 9	1.0%	0.7%
Harmonic 10	0.2%	0.0%
Harmonic 11	49.0%	49.0%
Harmonic 12	0.1%	0.0%
Harmonic 13	14.7%	15.4%
Harmonic 14	0.1%	0.0%
Harmonic 15	0.2%	0.1%
Harmonic 16	0.1%	0.0%
Harmonic 17	1.4%	2.3%
Harmonic 18	0.1%	0.0%
Harmonic 19	1.9%	1.6%
Harmonic 20	0.1%	0.0%
Harmonic 21	0.2%	0.3%
Harmonic 22	0.1%	0.0%
Harmonic 23	4.8%	5.1%
Harmonic 24	0.1%	0.1%

Harmonic 25	3.8%	3.9%
Harmonic 26	0.1%	0.1%
Harmonic 27	0.4%	0.5%
Harmonic 28	0.3%	0.4%
Harmonic 29	0.3%	0.4%
Harmonic 30	0.2%	0.2%
Harmonic 31	1.0%	1.5%
Harmonic 32	3.1%	4.0%
Harmonic 33	5.2%	6.4%
Harmonic 34	5.0%	5.8%
Harmonic 35	2.5%	2.9%
Harmonic 36	0.6%	0.6%
Harmonic 37	0.2%	0.2%
Harmonic 38	0.0%	0.0%
Harmonic 39	0.0%	0.0%
Harmonic 40	0.0%	0.0%
Harmonic 41	0.1%	0.1%
Harmonic 42	0.0%	0.0%
Harmonic 43	1.0%	0.8%
Harmonic 44	0.0%	0.0%
Harmonic 45	0.0%	0.0%
Harmonic 46	0.0%	0.0%
Harmonic 47	0.5%	0.3%
Harmonic 48	0.0%	0.0%
Harmonic 49	0.1%	0.0%
Harmonic 50	0.0%	0.0%
Harmonic 51	0.0%	0.0%
Harmonic 52	0.0%	0.0%
Harmonic 53	0.0%	0.0%
Harmonic 54	0.0%	0.0%
Harmonic 55	0.0%	0.0%
Harmonic 56	0.0%	0.0%
Harmonic 57	0.0%	0.0%
Harmonic 58	0.0%	0.0%
Harmonic 59	0.1%	0.0%
Harmonic 60	0.0%	0.0%
Harmonic 61	0.0%	0.0%
Harmonic 62	0.1%	0.0%
Harmonic 63	0.1%	0.2%
Harmonic 64	0.3%	0.5%
Harmonic 65	0.9%	1.2%
Harmonic 66	1.4%	1.8%
Harmonic 67	1.8%	2.1%

Harmonic 68	1.7%	2.1%
Harmonic 69	1.2%	1.5%
Harmonic 70	0.8%	0.8%
Harmonic 71	0.4%	0.3%
Harmonic 72	0.1%	0.1%
Harmonic 73	0.1%	0.1%
Harmonic 74	0.0%	0.0%
Harmonic 75	0.0%	0.0%
Harmonic 76	0.0%	0.0%
Harmonic 77	0.0%	0.1%
Harmonic 78	0.0%	0.0%
Harmonic 79	0.0%	0.0%
Harmonic 80	0.0%	0.0%
Harmonic 81	0.0%	0.0%
Harmonic 82	0.0%	0.0%
Harmonic 83	0.0%	0.0%
Harmonic 84	0.0%	0.0%
Harmonic 85	0.0%	0.0%
Harmonic 86	0.0%	0.0%
Harmonic 87	0.0%	0.0%
Harmonic 88	0.0%	0.0%
Harmonic 89	0.0%	0.0%
Harmonic 90	0.0%	0.0%
Harmonic 91	0.0%	0.1%
Harmonic 92	0.0%	0.0%
Harmonic 93	0.0%	0.0%
Harmonic 94	0.0%	0.1%
Harmonic 95	0.0%	0.1%
Harmonic 96	0.1%	0.0%
Harmonic 97 - 256	0.0%	0.0%

Table A.1: The percentage of non-zero values in Dataset 2, grouped by harmonic frequency and split between fault and non-fault samples.

Appendix B

Experiment 3 Results

Rank	Harmonic number	Time step feature	Line / Phase	Aggregation method	Importance
1	Harmonic 10	Minimum	V2	Maximum	0.00660
2	Harmonic 4	Minimum	V1	Standard deviation	0.00618
3	Harmonic 5	Minimum	V1	Standard deviation	0.00600
4	Harmonic 10	Minimum	V23	Maximum	0.00597
5	Harmonic 11	Minimum	V12	Standard deviation	0.00584
6	Harmonic 10	Minimum	V31	Maximum	0.00570
7	Harmonic 10	Minimum	V3	Maximum	0.00567
8	Harmonic 10	Minimum	V1	Maximum	0.00567
9	Harmonic 12	Minimum	V2	Maximum	0.00565
10	Harmonic 4	Minimum	V3	Standard deviation	0.00561
11	Harmonic 10	Minimum	V12	Standard deviation	0.00560
12	Harmonic 10	Minimum	V12	Maximum	0.00560
13	Harmonic 14	Minimum	V3	Standard deviation	0.00540
14	Harmonic 4	Minimum	V12	Standard deviation	0.00535
15	Harmonic 4	Minimum	V2	Standard deviation	0.00534
16	Harmonic 14	Minimum	V1	Standard deviation	0.00530
17	Harmonic 10	Minimum	V12	Mean	0.00529
18	Harmonic 15	Minimum	V12	Standard deviation	0.00528
19	Harmonic 15	Minimum	V23	Standard deviation	0.00527
20	Harmonic 15	Minimum	V3	Standard deviation	0.00525
21	Harmonic 4	Minimum	V31	Standard deviation	0.00523
22	Harmonic 11	Minimum	V23	Maximum	0.00519
23	Harmonic 14	Minimum	V23	Standard deviation	0.00519
24	Harmonic 5	Minimum	V31	Standard deviation	0.00517

25	Harmonic 5	Minimum	V3	Standard deviation	0.00513
26	Harmonic 10	Minimum	V23	Standard deviation	0.00512
27	Harmonic 5	Minimum	V2	Standard deviation	0.00510
28	Harmonic 15	Minimum	V1	Standard deviation	0.00510
29	Harmonic 10	Minimum	V2	Standard deviation	0.00508
30	Harmonic 5	Minimum	V12	Standard deviation	0.00508
31	Harmonic 4	Minimum	V23	Standard deviation	0.00507
32	Harmonic 11	Minimum	V23	Standard deviation	0.00507
33	Harmonic 11	Minimum	V23	Mean	0.00504
34	Harmonic 10	Minimum	V2	Mean	0.00504
35	Harmonic 11	Minimum	V31	Standard deviation	0.00503
36	Harmonic 5	Minimum	V23	Standard deviation	0.00503
37	Harmonic 10	Minimum	V31	Standard deviation	0.00503
38	Harmonic 14	Minimum	V31	Maximum	0.00502
39	Harmonic 11	Minimum	V2	Standard deviation	0.00501
40	Harmonic 10	Minimum	V31	Mean	0.00501
41	Harmonic 11	Minimum	V2	Maximum	0.00499
42	Harmonic 14	Minimum	V1	Maximum	0.00492
43	Harmonic 15	Minimum	V2	Standard deviation	0.00488
44	Harmonic 11	Minimum	V12	Mean	0.00488
45	Harmonic 14	Minimum	V3	Maximum	0.00486
46	Harmonic 15	Minimum	V31	Standard deviation	0.00485
47	Harmonic 10	Minimum	V1	Standard deviation	0.00484
48	Harmonic 14	Minimum	V31	Standard deviation	0.00484
49	Harmonic 14	Minimum	V2	Maximum	0.00482
50	Harmonic 10	Minimum	V23	Mean	0.00482
51	Harmonic 4	Minimum	V23	Maximum	0.00482
52	Harmonic 11	Minimum	V12	Maximum	0.00481
53	Harmonic 14	Minimum	V12	Standard deviation	0.00478
54	Harmonic 4	Minimum	V2	Maximum	0.00474
55	Harmonic 14	Minimum	V2	Standard deviation	0.00472
56	Harmonic 10	Minimum	V3	Standard deviation	0.00471
57	Harmonic 14	Minimum	V12	Maximum	0.00471
58	Harmonic 11	Minimum	V2	Mean	0.00468
59	Harmonic 4	Minimum	V1	Maximum	0.00467
60	Harmonic 14	Minimum	V31	Mean	0.00466
61	Harmonic 11	Minimum	V31	Maximum	0.00460
62	Harmonic 4	Minimum	V31	Maximum	0.00455
63	Harmonic 5	Minimum	V1	Mean	0.00454
64	Harmonic 5	Minimum	V23	Mean	0.00453
65	Harmonic 4	Minimum	V3	Maximum	0.00451
66	Harmonic 15	Minimum	V2	Maximum	0.00451
67	Harmonic 5	Minimum	V31	Maximum	0.00450

68	Harmonic 12	Minimum	V1	Maximum	0.00449
69	Harmonic 15	Minimum	V23	Mean	0.00449
70	Harmonic 4	Minimum	V12	Mean	0.00448
71	Harmonic 4	Minimum	V12	Maximum	0.00445
72	Harmonic 14	Minimum	V23	Mean	0.00443
73	Harmonic 5	Minimum	V3	Maximum	0.00441
74	Harmonic 4	Minimum	V1	Mean	0.00441
75	Harmonic 15	Minimum	V3	Maximum	0.00438
76	Harmonic 15	Minimum	V23	Maximum	0.00437
77	Harmonic 14	Minimum	V2	Mean	0.00436
78	Harmonic 5	Minimum	V1	Maximum	0.00436
79	Harmonic 11	Minimum	V1	Standard deviation	0.00435
80	Harmonic 5	Minimum	V31	Mean	0.00434
81	Harmonic 5	Minimum	V3	Mean	0.00433
82	Harmonic 5	Minimum	V2	Mean	0.00432
83	Harmonic 5	Minimum	V23	Maximum	0.00431
84	Harmonic 2	Minimum	V2	Maximum	0.00431
85	Harmonic 15	Minimum	V2	Mean	0.00429
86	Harmonic 14	Minimum	V23	Maximum	0.00429
87	Harmonic 10	Minimum	V1	Mean	0.00427
88	Harmonic 4	Minimum	V3	Mean	0.00426
89	Harmonic 11	Minimum	V31	Mean	0.00425
90	Harmonic 14	Minimum	V3	Mean	0.00425
91	Harmonic 15	Minimum	V12	Maximum	0.00424
92	Harmonic 15	Minimum	V12	Mean	0.00421
93	Harmonic 11	Minimum	V3	Mean	0.00421
94	Harmonic 5	Minimum	V12	Mean	0.00419
95	Harmonic 15	Minimum	V3	Mean	0.00418
96	Harmonic 10	Minimum	V3	Mean	0.00417
97	Harmonic 14	Minimum	V12	Mean	0.00416
98	Harmonic 4	Minimum	V31	Mean	0.00416
99	Harmonic 15	Minimum	V31	Mean	0.00415
100	Harmonic 11	Minimum	V3	Maximum	0.00414
101	Harmonic 15	Minimum	V1	Mean	0.00413
102	Harmonic 5	Minimum	V2	Maximum	0.00408
103	Harmonic 4	Minimum	V23	Mean	0.00407
104	Harmonic 11	Minimum	V1	Maximum	0.00405
105	Harmonic 15	Minimum	V1	Maximum	0.00399
106	Harmonic 15	Minimum	V31	Maximum	0.00398
107	Harmonic 11	Minimum	V3	Standard deviation	0.00394
108	Harmonic 12	Minimum	V3	Maximum	0.00393
109	Harmonic 5	Minimum	V12	Maximum	0.00393
110	Harmonic 14	Minimum	V1	Mean	0.00391

111	Harmonic 11	Minimum	V1	Mean	0.00385
112	Harmonic 4	Minimum	V2	Mean	0.00382
113	Harmonic 15	Maximum	V31	Standard deviation	0.00375
114	Harmonic 2	Minimum	V1	Maximum	0.00373
115	Harmonic 14	Maximum	V23	Standard deviation	0.00371
116	Harmonic 5	Minimum	V1	Minimum	0.00370
117	Harmonic 5	Minimum	V2	Minimum	0.00366
118	Harmonic 4	Minimum	V1	Minimum	0.00360
119	Harmonic 14	Maximum	V3	Standard deviation	0.00357
120	Harmonic 15	Maximum	V3	Standard deviation	0.00354

Table B.1: The 120 most important features in Dataset 3, ranked by a Random Forest in Experiment 3.

Appendix C

Features used in Experiment 4

Harmonic number	Time step feature	Line / Phase	Aggregation method
Harmonic 2	Minimum	V1	Maximum
Harmonic 2	Minimum	V2	Maximum
Harmonic 2	Minimum	V3	Maximum
Harmonic 2	Minimum	V12	Maximum
Harmonic 2	Minimum	V23	Maximum
Harmonic 2	Minimum	V31	Maximum
Harmonic 3	Minimum	V1	Maximum
Harmonic 3	Minimum	V2	Maximum
Harmonic 3	Minimum	V3	Maximum
Harmonic 3	Minimum	V12	Maximum
Harmonic 3	Minimum	V23	Maximum
Harmonic 3	Minimum	V31	Maximum
Harmonic 4	Minimum	V1	Maximum
Harmonic 4	Minimum	V2	Maximum
Harmonic 4	Minimum	V3	Maximum
Harmonic 4	Minimum	V12	Maximum
Harmonic 4	Minimum	V23	Maximum
Harmonic 4	Minimum	V31	Maximum
Harmonic 5	Minimum	V1	Maximum
Harmonic 5	Minimum	V2	Maximum
Harmonic 5	Minimum	V3	Maximum
Harmonic 5	Minimum	V12	Maximum
Harmonic 5	Minimum	V23	Maximum
Harmonic 5	Minimum	V31	Maximum
Harmonic 6	Minimum	V1	Maximum

Harmonic 6	Minimum	V2	Maximum
Harmonic 6	Minimum	V3	Maximum
Harmonic 6	Minimum	V12	Maximum
Harmonic 6	Minimum	V23	Maximum
Harmonic 6	Minimum	V31	Maximum
Harmonic 7	Minimum	V1	Maximum
Harmonic 7	Minimum	V2	Maximum
Harmonic 7	Minimum	V3	Maximum
Harmonic 7	Minimum	V12	Maximum
Harmonic 7	Minimum	V23	Maximum
Harmonic 7	Minimum	V31	Maximum
Harmonic 9	Minimum	V1	Maximum
Harmonic 9	Minimum	V2	Maximum
Harmonic 9	Minimum	V3	Maximum
Harmonic 9	Minimum	V12	Maximum
Harmonic 9	Minimum	V23	Maximum
Harmonic 9	Minimum	V31	Maximum
Harmonic 11	Minimum	V1	Maximum
Harmonic 11	Minimum	V2	Maximum
Harmonic 11	Minimum	V3	Maximum
Harmonic 11	Minimum	V12	Maximum
Harmonic 11	Minimum	V23	Maximum
Harmonic 11	Minimum	V31	Maximum
Harmonic 13	Minimum	V1	Maximum
Harmonic 13	Minimum	V2	Maximum
Harmonic 13	Minimum	V3	Maximum
Harmonic 13	Minimum	V12	Maximum
Harmonic 13	Minimum	V23	Maximum
Harmonic 13	Minimum	V31	Maximum
Harmonic 2	Minimum	V1	Mean
Harmonic 2	Minimum	V2	Mean
Harmonic 2	Minimum	V3	Mean
Harmonic 2	Minimum	V12	Mean
Harmonic 2	Minimum	V23	Mean
Harmonic 2	Minimum	V31	Mean
Harmonic 3	Minimum	V1	Mean
Harmonic 3	Minimum	V2	Mean
Harmonic 3	Minimum	V3	Mean
Harmonic 3	Minimum	V12	Mean
Harmonic 3	Minimum	V23	Mean
Harmonic 3	Minimum	V31	Mean
Harmonic 4	Minimum	V1	Mean
Harmonic 4	Minimum	V2	Mean

Harmonic 4	Minimum	V3	Mean
Harmonic 4	Minimum	V12	Mean
Harmonic 4	Minimum	V23	Mean
Harmonic 4	Minimum	V31	Mean
Harmonic 5	Minimum	V1	Mean
Harmonic 5	Minimum	V2	Mean
Harmonic 5	Minimum	V3	Mean
Harmonic 5	Minimum	V12	Mean
Harmonic 5	Minimum	V23	Mean
Harmonic 5	Minimum	V31	Mean
Harmonic 6	Minimum	V1	Mean
Harmonic 6	Minimum	V2	Mean
Harmonic 6	Minimum	V3	Mean
Harmonic 6	Minimum	V12	Mean
Harmonic 6	Minimum	V23	Mean
Harmonic 6	Minimum	V31	Mean
Harmonic 7	Minimum	V1	Mean
Harmonic 7	Minimum	V2	Mean
Harmonic 7	Minimum	V3	Mean
Harmonic 7	Minimum	V12	Mean
Harmonic 7	Minimum	V23	Mean
Harmonic 7	Minimum	V31	Mean
Harmonic 9	Minimum	V1	Mean
Harmonic 9	Minimum	V2	Mean
Harmonic 9	Minimum	V3	Mean
Harmonic 9	Minimum	V12	Mean
Harmonic 9	Minimum	V23	Mean
Harmonic 9	Minimum	V31	Mean
Harmonic 11	Minimum	V1	Mean
Harmonic 11	Minimum	V2	Mean
Harmonic 11	Minimum	V3	Mean
Harmonic 11	Minimum	V12	Mean
Harmonic 11	Minimum	V23	Mean
Harmonic 11	Minimum	V31	Mean
Harmonic 13	Minimum	V1	Mean
Harmonic 13	Minimum	V2	Mean
Harmonic 13	Minimum	V3	Mean
Harmonic 13	Minimum	V12	Mean
Harmonic 13	Minimum	V23	Mean
Harmonic 13	Minimum	V31	Mean
Harmonic 2	Minimum	V1	Standard deviation
Harmonic 2	Minimum	V2	Standard deviation
Harmonic 2	Minimum	V3	Standard deviation

Harmonic 2	Minimum	V12	Standard deviation
Harmonic 2	Minimum	V23	Standard deviation
Harmonic 2	Minimum	V31	Standard deviation
Harmonic 3	Minimum	V1	Standard deviation
Harmonic 3	Minimum	V2	Standard deviation
Harmonic 3	Minimum	V3	Standard deviation
Harmonic 3	Minimum	V12	Standard deviation
Harmonic 3	Minimum	V23	Standard deviation
Harmonic 3	Minimum	V31	Standard deviation
Harmonic 4	Minimum	V1	Standard deviation
Harmonic 4	Minimum	V2	Standard deviation
Harmonic 4	Minimum	V3	Standard deviation
Harmonic 4	Minimum	V12	Standard deviation
Harmonic 4	Minimum	V23	Standard deviation
Harmonic 4	Minimum	V31	Standard deviation
Harmonic 5	Minimum	V1	Standard deviation
Harmonic 5	Minimum	V2	Standard deviation
Harmonic 5	Minimum	V3	Standard deviation
Harmonic 5	Minimum	V12	Standard deviation
Harmonic 5	Minimum	V23	Standard deviation
Harmonic 5	Minimum	V31	Standard deviation
Harmonic 6	Minimum	V1	Standard deviation
Harmonic 6	Minimum	V2	Standard deviation
Harmonic 6	Minimum	V3	Standard deviation
Harmonic 6	Minimum	V12	Standard deviation
Harmonic 6	Minimum	V23	Standard deviation
Harmonic 6	Minimum	V31	Standard deviation
Harmonic 7	Minimum	V1	Standard deviation
Harmonic 7	Minimum	V2	Standard deviation
Harmonic 7	Minimum	V3	Standard deviation
Harmonic 7	Minimum	V12	Standard deviation
Harmonic 7	Minimum	V23	Standard deviation
Harmonic 7	Minimum	V31	Standard deviation
Harmonic 9	Minimum	V1	Standard deviation
Harmonic 9	Minimum	V2	Standard deviation
Harmonic 9	Minimum	V3	Standard deviation
Harmonic 9	Minimum	V12	Standard deviation
Harmonic 9	Minimum	V23	Standard deviation
Harmonic 9	Minimum	V31	Standard deviation
Harmonic 11	Minimum	V1	Standard deviation
Harmonic 11	Minimum	V2	Standard deviation
Harmonic 11	Minimum	V3	Standard deviation
Harmonic 11	Minimum	V12	Standard deviation

Harmonic 11	Minimum	V23	Standard deviation
Harmonic 11	Minimum	V31	Standard deviation
Harmonic 13	Minimum	V1	Standard deviation
Harmonic 13	Minimum	V2	Standard deviation
Harmonic 13	Minimum	V3	Standard deviation
Harmonic 13	Minimum	V12	Standard deviation
Harmonic 13	Minimum	V23	Standard deviation
Harmonic 13	Minimum	V31	Standard deviation

Table C.1: The features used in Experiment 4.

Appendix D

Experiment 4 Results

Kernel	Error penalty multiplier	Mean accuracy	Mean MCC
Linear	1	0.5891	0.2339
Sigmoid	1	0.5819	0.1636
Radial basis function	1	0.5778	0.2433
Linear	10	0.5926	0.2225
Sigmoid	10	0.5744	0.1487
Radial basis function	10	0.5858	0.2437
2^{nd} degree polynomial	1	0.5325	0.1597
3^{rd} degree polynomial	1	0.5292	0.1537
4^{th} degree polynomial	1	0.5276	0.1481
2^{nd} degree polynomial	10	0.5407	0.1788
3^{rd} degree polynomial	10	0.5329	0.1636
4^{th} degree polynomial	10	0.5309	0.1598

Table D.1: Accuracy and MCC scores for the SVMs tested in Experiment 4.

Number of trees	Max depth	Mean accuracy	Mean MCC
100	20	0.7286	0.4669
250	20	0.7265	0.4627
350	20	0.7265	0.4620
500	20	0.7263	0.4618

400	20	0.7259	0.4611
450	20	0.7257	0.4608
200	20	0.7254	0.4607
300	20	0.7246	0.4585
150	20	0.7241	0.4578
50	20	0.7249	0.4578
400	25	0.7256	0.4570
400	None	0.7263	0.4568
250	None	0.7259	0.4561
500	None	0.7256	0.4555
300	None	0.7254	0.4551
350	25	0.7243	0.4545
500	25	0.7241	0.4540
150	None	0.7246	0.4536
200	None	0.7244	0.4532
350	None	0.7246	0.4532
450	None	0.7244	0.4530
150	25	0.7231	0.4524
300	25	0.7228	0.4516
100	25	0.7231	0.4515
450	25	0.7220	0.4495
250	25	0.7202	0.4467
100	None	0.7210	0.4467
200	25	0.7189	0.4441
350	15	0.7111	0.4416
500	15	0.7109	0.4405
50	None	0.7174	0.4401
450	15	0.7103	0.4393
300	15	0.7097	0.4384
400	15	0.7087	0.4362
50	25	0.7156	0.4360
200	15	0.7085	0.4357
150	15	0.7081	0.4356
250	15	0.7080	0.4343
100	15	0.7057	0.4300
50	15	0.7025	0.4209
500	10	0.6730	0.3783
250	10	0.6730	0.3773
400	10	0.6722	0.3764
450	10	0.6720	0.3763
350	10	0.6717	0.3752
300	10	0.6714	0.3745
100	10	0.6709	0.3710

200	10	0.6701	0.3705
150	10	0.6681	0.3663
50	10	0.6687	0.3636
300	5	0.6219	0.2675
250	5	0.6215	0.2669
450	5	0.6216	0.2668
350	5	0.6209	0.2655
400	5	0.6209	0.2652
150	5	0.6206	0.2651
100	5	0.6202	0.2643
500	5	0.6203	0.2638
200	5	0.6200	0.2635
50	5	0.6196	0.2620

Table D.2: Accuracy and MCC scores for the Random Forests tested in Experiment 4

First layer	Second layer	Third layer	Mean accuracy	Mean MCC
64	32	32	0.6765	0.3498
64	32	64	0.6921	0.3824
64	64	32	0.6765	0.3503
64	64	64	0.6852	0.3665
128	32	32	0.6852	0.3690
128	32	64	0.6892	0.3759
128	64	32	0.6927	0.3821
128	64	64	0.6996	0.3960
128	64	128	0.6632	0.3217
128	128	64	0.6736	0.3459
128	128	128	0.6580	0.3165
256	64	64	0.6707	0.3420
256	64	128	0.6840	0.3654
256	128	64	0.6701	0.3359
256	128	128	0.6753	0.3478

Table D.3: Accuracy and MCC scores for the Feed-Forward Neural Networks tested in Experiment 4.

First bidirectional layer size	Second bidirectional layer size	Fully-connected layer size	Mean accuracy	Mean MCC
32	None	16	0.6482	0.2949
32	None	32	0.6701	0.3374
32	32	16	0.6545	0.3063

32	32	32	0.6736	0.3455
64	None	16	0.6661	0.3281
64	None	32	0.6759	0.3495
64	32	16	0.6701	0.3382
64	32	32	0.6979	0.3925
64	32	64	0.6701	0.3370
64	64	32	0.6771	0.3521
64	64	64	0.6771	0.3529
128	32	32	0.6852	0.3691
128	32	64	0.6782	0.3540
128	64	32	0.6805	0.3615
128	64	64	0.6840	0.3661
128	64	64	0.6909	0.3796
128	64	128	0.6892	0.3774
128	128	64	0.6852	0.3674
128	128	128	0.6834	0.3656

Table D.4: Accuracy and MCC scores for the Recurrent Neural Networks tested in Experiment 4.

Vemund Mehl Santi

Predicting faults in power grids using machine learning methods

NTNU
Norwegian University of
Science and Technology

SINTEF