Norwegian University
of Life Sciences

Master's Thesis 2021 30 ECTS
Faculty of Science and Technology

A Study of Machine Learning Methods Used as Decision Support Tool for Grid Operators – The NEWEPS Project

Tobias Korten
Environmental Physics and Renewable Energy

The truth may be puzzling. It may take some work to grapple with.
It may be counterintuitive. It may contradict deeply held prejudices.
It may not be consonant with what we desperately want to be true.
But our preferences do not determine what's true.

Carl Sagan
American Astronomer

Acknowledgements

This master's thesis marks the completion of my five-year master's degree in Environmental Physics and Renewable Energy at the Norwegian University of Life Sciences (NMBU). The last five years have been filled with exciting experiences, thanks to a wonderful student environment and inspiring and enthusiastic professors. I will be forever grateful for the memories and friendships that have been created during my time at NMBU.

I would like to thank my supervisor Sonja Monica Berlijn for giving me the opportunity to be part of the NEWEPS project and for all the help, expertise, and guidance through this process. I also want to thank the Statnett employees that were involved in this thesis, for taking their time to help me gather information and answering my questions. In addition, I want to express gratitude to my fellow students working on this project: Krishna Solberg, Ellen Bera Mathiesen, and Andreas Svanes. Without you, this writing process would not have been the same enjoyable experience as it has been. Thank you so much for the continuous support, interesting discussions, and encouraging words which motivated me to finalize this thesis. Finally, I would like to thank my girlfriend and family, for supporting me from beginning to end, and for giving me valuable feedback through proofreading.

Working on this master's thesis has been an interesting, challenging, and valuable experience. The thesis themes are highly relevant for the future power system. I hope it will be of interest to Statnett and other stakeholders in the power system.

Ås, May 2021

Tobias Korten

Abstract

The electrical power system is becoming increasingly dynamic and complex. Through the Green Deal, the European Union (EU) aims at decarbonizing the energy sector, shifting from fossil fuels to renewable energy sources. This trend gives rise to unreliable power generation and unpredictable consumer patterns with larger loads. Combined with more frequent extreme weather conditions due to global warming more faults, disturbances, and stability issues in the power grid are predicted to happen. To prepare for this future, the Nordic Transmission System Operators (TSOs) have initiated the Nordic Early Warning Early Prevention System (NEWEPS) project. The project aims at creating a system that acts as a decision support tool for grid operators. The goal is to create a system that can give warnings about coming faults and disturbances, and stability issues in the grid before they appear. As this system will use data from Phasor Measurement Units (PMUs) as an information source, large amounts of data need to be analyzed in real-time. A data analysis method that could handle such large amounts of data is Machine Learning (ML).

This master's thesis is a contribution to this project, as it will explore the possibility of using state-of-the-art ML models to predict faults and stability in the power grid. A literature review and a case study were performed. The literature search resulted in 16 articles that met the set limitations. These articles were categorized as either predicting power system stability in the post-fault timeframe or predicting faults and disturbances in the pre-fault timeframe. This produced a valuable overview of the most promising predictive ML models currently researched. The discussion resulted in a recommendation of the most relevant models for each category. For the post-fault timeframe, a Recurrent Neural Network (RNN) with Long Short-Term Memory (LSTM) and a Feed-Forward Neural Network (FFNN) were recommended and for the pre-fault timeframe, a hybrid model using RNN with LSTM and a Support Vector Machine (SVM) classifier was recommended. The case study illustrates which patterns ML models can learn and detect to give accurate predictions. It is also found that some minor faults produce subtle anomalies in the waveforms before they evolve into major faults. These anomalies, oftentimes unnoticed by human grid operators, could be detected using trained ML models.

Based on the results from the literature review, a Technology Readiness Level (TRL) evaluation of each category is carried out. Both categories have been evaluated to be between a TRL 3 and 4. A TRL evaluation of the current usage of such models by the Nordic TSOs is also performed. These TRLs are compared, and it is found that the research done on this topic by the Nordic TSOs is very limited. The results from this master's thesis could therefore prove to be a great starting point for this research. Based on the literature study and the case study done in this thesis it has been shown that predictive ML models can increase awareness in the power grid and assist grid operators in power grid management, by providing early warnings about faults and stability.

Sammendrag

Det elektriske kraftsystemet blir stadig mer dynamisk og komplekst. Gjennom Green Deal har EU som mål å avkarbonisere energisektoren, og skifte fra fossile brensler til fornybare energikilder. Denne trenden gir opphav til upålitelig kraftproduksjon og uforutsigbare forbrukermønstre med større belastning. Kombinert med hyppigere ekstreme værforhold på grunn av global oppvarming, forventes det at det i fremtiden vil oppstå flere feil, forstyrrelser og stabilitetsutfordringer i strømnettet. For å forberede seg på denne fremtiden har de nordiske transmisjonsnettoperatørene (TSOene) startet prosjektet Nordic Early Warning Early Prevention System (NEWEPS). Prosjektet har som mål å skape et system som fungerer som et beslutningsstøtteverktøy for nettoperatører. Målet er å lage et system som kan gi advarsler om kommende feil og forstyrrelser, og stabilitetsproblemer i nettet før de inntreffer. Ettersom dette systemet vil bruke data fra Phasor Measurement Units (PMUs) som informasjonskilde, må store datamengder analyseres i sanntid. En dataanalysemetode som kan håndtere så store datamengder er Maskinlæring (ML).

Denne masteroppgaven er et bidrag til dette prosjektet, da det vil utforske muligheten for å bruke state-of-the-art ML-modeller for å forutsi feil og stabilitet i strømnettet. Det ble utført en litteraturstudie og en casestudie. Litteratursøket resulterte i 16 artikler som oppfylte de angitte begrensningene. Disse artiklene er kategorisert til å enten forutsi nettstabilitet i tidsrammen rett etter at en feil har inntruffet eller forutsi feil og forstyrrelser i tidsrammen før feilen har inntruffet. En oversikt over de mets lovende prediktive ML-modellene har blitt presentert i form av tabeller. Diskusjonen resulterte i en anbefaling av de mest relevante modellene for hver kategori. I den første kategorien ble en RNN med LSTM og en FFNN anbefalt, og for den andre kategorien ble en hybridmodell basert på en RNN med LSTM og en SVM anbefalt. Casestudien illustrerer hvilke mønstre ML-modeller kan lære seg og oppdage for å gi nøyaktige prediksjoner. Det har også blitt funnet at noen feil produserer subtile uregelmessigheter i bølgeformene før de utvikler seg til større feil. Disse avvikene, ofte ubemerket av menneskelige nettoperatører, kan oppdages ved hjelp av trente ML-modeller.

Basert på resultatene fra litteraturstudien, utføres en TRL-evaluering (Teknologimodenhet) av hver kategori. Begge kategoriene er evaluert til å være mellom TRL 3 og 4. En TRL-evaluering av dagens bruk av slike modeller av de nordiske TSO-ene blir også utført. Disse TRL-ene blir sammenlignet, og det er funnet at forskningen som er gjort på dette emnet av de nordiske TSO-ene er svært begrenset. Resultatene fra denne masteroppgaven kan derfor vise seg å være et godt utgangspunkt for denne forskningen. Basert på litteraturstudien og casestudien gjennomført i denne masteroppgaven har det blitt vist at prediktive ML-modeller har evnen til å øke oversikten og kontrollen i kraftnettet, og dermed hjelpe nettoperatører med kraftnettadministrasjon ved å gi tidlige varsler om feil og stabilitet.

Table of Contents

Abbreviations

ANN	Artificial Neural Network
DT	Decision Tree
DFA	Distribution Fault Anticipator
ENTSO-E	European Network of Transmission System Operators for Electricity
EU	European Union
FN	False Negative
FP	False Positive
FFNN	Feed-Forward Neural Network
GPS	Global Positioning System
HVDC	High-Voltage Direct Current
LR	Linear Regression
LSTM	Long Short-Term Memory
LFO	Low-Frequency Oscillation
ML	Machine Learning
MSE	Mean Squared Error
MLP	Multilayer Perceptron
NEWEPS	Nordic Early Warning Early Prevention System
NTNU	Norwegian University of Science and Technology
OLS	Ordinary Least Squares
PMU	Phasor Measurement Unit
PQA	Power Quality Analyzer
PCA	Principal Component Analysis
RBF	Radial Basis Function
RBFNN	Radial Basis Function Neural Network
RF	Random Forest
RNN	Recurrent Neural Network
RTU	Remote Terminal Unit
RMS	Root Mean Squared
RMSE	Root Mean Squared Error
SCADA	Supervisory control and data acquisition
SVM	Support Vector Machine
SVR	Support Vector Regressor
TRL	Technology Readiness Level
TSO	Transmission System Operator
TN	True Negative
TP	True Positive
UE	Undelivered Energy

Chapter 1: Introduction

1.1 Background and motivation

Modern society is becoming more and more dependent on energy. On a global level, energy consumption has been steadily increasing since the industrial revolution [1]. At that time, the use of fossil fuels such as coal and oil sparked innovation and new technologies. Starting with the steam turbine in the late 1800s, many new technologies have been invented, increasing the standards of living. In recent years, the negative effects of the industrial revolution and the use of fossil fuels have become apparent. The emissions of greenhouse gases in the atmosphere have led to a greenhouse effect, which in turn has given rise to more extreme weather and changing weather patterns, increased sea levels, and global temperatures [2].

In an attempt to reduce the adverse impacts of the greenhouse effect, many projects and incentives have been proposed to reduce the emission of greenhouse gases. The European Green Deal is such a project. Initiated by the European Union (EU) at the end of 2019, the project's goal is to make Europa the first climate-neutral continent within the year 2050. The EU is planning to achieve this by investing in environmentally friendly technologies and by decarbonizing the energy sector, both in power generation, transmission, and end-user power consumption [3]. To achieve this, electrification will have to play a major role. Electrification is here meant by replacing direct use of energy from fossil fuels with renewable energy.

This progression towards electrification and decarbonizing in the EU challenges the Transmission System Operators (TSOs) to maintain a reliable power grid, as this shift will lead to changes in both the production and consumer end of the grid, which in turn introduces low inertia and frequency instabilities and makes the grid more complex. This will make grid operation more difficult and increases the chances of faults and disturbances. For this reason, a lot of research on smart grids has been done in recent years [4].

In smart grids, communication technology and sensors play a central role. One such sensor is the Phasor Measurement Unit (PMU). These sensors measure the magnitude and the phase angle of voltage and current at a high sampling rate [5]. This results in huge amounts of data, reaching 310 GB of raw data per day for a PMU system consisting of 1100 units [6]. This has encouraged TSOs to turn to Machine Learning (ML) methods for evaluating these vast amounts of data. ML is especially suited for such tasks, as it can analyze, find patterns, and learn from large datasets. This makes them able to predict future events based on past examples [7]. These models can help grid operators to keep the system secure, by giving early warnings about stability and upcoming faults in the grid and assisting in decision making [8]. For that reason, a lot of articles have been written about the usage of ML in power grid operation in the last two decades. A distinction between predictive and non-predictive models should be made. Non-predictive models are associated with identifying and locating faults as they happen. Predictive models forecast upcoming faults and disturbances before they take place. Such models can also be used for predicting power system stability right after a fault or disturbance in the grid has happened [8].

1.2 Scope and limitations

In this master's thesis, the goal is to assess predictive, state-of-the-art ML algorithms that are being used for fault applications in electrical power systems. A literature review is performed, where the articles containing these predictive ML models are categorized based on the type of fault that the article is concerned with. As this master's thesis is a contribution to the Nordic TSOs project, Nordic Early Warning Early Prevention System (NEWEPS) [9], the project is in focus when assessing the ML models. Research on the most recent ML models is emphasized in this thesis as this is most relevant for this project. Because the upcoming system will use PMU data, ML models using data from PMUs have been in focus, and articles associated with PMU based ML models have, to the degree it was possible, been chosen. More information on the project and PMU will be presented later in the thesis.

A Technology Readiness Level (TRL) assessment of each category of predictive ML models is performed. The relevance of each category for the NEWEPS project is determined. Based on the literature review, a recommendation for the NEWEPS project on which areas to focus on will be given. Further limitations for the literature review are specified in Chapter 5.

As this master's thesis will be a contribution to the NEWEPS project, the focus will be on the Nordic synchronous power system. This consist of the following four TSOs: Statnett SF (Norway), Svenska Kräftnet (Sweden), Energinet (Denmark) and Fingrid (Finland).

This master's thesis was completed in four months. The literature review study and the case study were chosen as the most appropriate methods. The literature study gives a comprehensive overview of the state-of-the-art ML models. This overview can be used as a reference point for further research on this topic. The case study displays how these models can be used in real-time power grid operation. In this thesis, coding and testing of the ML models presented in the literature study were not prioritized. This thesis rather lays the foundation for future participants of the NEWEPS project to implement the recommended models.

1.3 Research question

In this thesis, the following research question will be answered:

> *How can predictive, PMU based, state-of-the-art Machine Learning models be used as real-time decision support for grid operators in both pre- and post-fault events?*

This will be determined based on some criteria. Speed and performance are two important criteria for an ML model that is going to be used in real-time grid operation. The methods each author has used to train and test each algorithm will also be discussed. The research question gives rise to the following sub-questions:

- *To what extend is it possible to use Machine Learning models to predict the stability of the grid immediately after a fault happens?*
- *To what extend is it possible to use Machine Learning models to predict upcoming faults in the grid before the fault happens?*
- *How relevant are the predictive Machine Learning models for Statnett and the NEWEPS project?*

Answering these sub-questions will give a more comprehensive answer to the research question.

1.4 Structure of this master's thesis

In the first chapter, some fundamental knowledge about the power system will be presented. Chapter 2 explains the current trends in the Nordic power system considering the shift towards green technology in the EU. These trends introduce challenges for the Nordic TSOs which will be presented. The NEWEPS project is introduced as a response to these upcoming challenges. PMUs and PMU technology will be explained in the succeeding chapter, followed by a description of the research methods used in this thesis in Chapter 5. In Chapter 6, a thorough presentation of ML theory will be given, including examples of relevant ML algorithms that are used in the literature. In Chapter 7, the pre-fault and post-fault categories are explained and defined. In the following chapter, the results from the literature search are presented and the methods used in each article are discussed. Chapter 9 contains information about the case study along with the results from the case study. In Chapter 10, the results from the literature review are discussed in consideration of the NEWEPS project and TRL evaluations are performed. Recommendations for the project will also be given here. A conclusion on the thesis will be given in Chapter 11.

Chapter 2: Fundamentals about the power system

2.1 The electrical power system

An electrical power system has mainly three functions [10]:

- Production: Either through power plants, producing electricity by using energy sources, such as coal, gas, and wind, to convert mechanical energy to electrical energy, or through import from other countries.
- Transmission: Through power lines, cables, and transformers. Power lines and cables are used to distribute the electric power from the production facilities to the consumers, while transformers are used to transform the voltage of the electric power either up or down.
- Consumption: Either through end-user consumption such as industries or households, using the electric power, or through export to other countries.

2.1.1 The Norwegian grid

The electrical grid in Norway consists of three grid levels: the transmission grid, the regional grid, and the distribution grid. The transmission grid, delivering electricity at a voltage level of either 132 kV, 300 kV, or 420 kV, is operated by Statnett. This grid transports the power over long distances, not only to the regional grids in Norway but also to other countries. Between the distribution and the regional grid, there is a step-down transformer that transforms the voltage down to a voltage level of 33-132 kV. The regional grid mainly works as a link between the high voltage transmission grid and the low-voltage distribution grid. Large, power-intensive industries, however, connect directly to this grid. The distribution grid, at voltage level ≤ 22 kV, is the grid to which small-end consumers are connected to. Both the regional grid and the distribution grid are operated by approximately 130 distribution system operators in Norway [10].

2.2 Frequency

Besides maintaining the transmission grid, the biggest responsibility for a TSO is to ensure that the instantaneous balance between production and consumption is preserved, visualized by the following equation:

$$production + import = consumption + export + losses \tag{1}$$

As can be seen from the equation, import and export of electrical energy with other countries is also an important part of this balance. Eq. 1 says that a change in consumption must immediately be followed by a change in production. To give a simple illustration: when a light bulb is turned on, the power needed to light this bulb has to simultaneously be produced at a power plant.

The system's frequency is a measure of this equation. If there is more production than consumption in the system, the frequency will increase. Conversely, if there is more consumption than production, the frequency will decrease. In the Nordic power system, the frequency should be as close to 50 Hz as possible, not deviating more than 0,1 Hz [11]. If the frequency deviates too much, counteractions must be taken by the TSO to establish a balance. Counteractions could be decreasing or increasing energy production, reducing consumption, importing or exporting, and, as a last resort, load and production shedding. When load shedding, the TSO deprives a chosen set of electrical consumers of power supply. If counteractions are not done in time, power plants may start to switch off. In the worst-case scenario, this might end in a blackout. Keeping a stable 50 Hz grid requires careful monitoring of the grid and reliable predictions of system stability and upcoming fault events.

2.3 The Nordic power system

The Nordic power system is part of the European Network of Transmission System Operators for Electricity (ENTSO-E). A map of the five synchronous areas that are members of the ENTSOE-E is given in Figure 1. The power systems in each synchronous area are connected and are operating at the same frequency [12]. This means that a change in production or consumption in one country of the area will affect all the other countries in the same area. The task of keeping the frequency at 50 Hz, therefore, requires constant communication and cooperation between the countries in the same synchronous area.

Although not synchronously connected, the Nordic synchronous area is connected to the other synchronous areas through High-Voltage Direct Current (HVDC) interconnectors. These interconnections are of high interest for the Nordic power system, as they can help balance Eq. 1 by exporting and importing power.

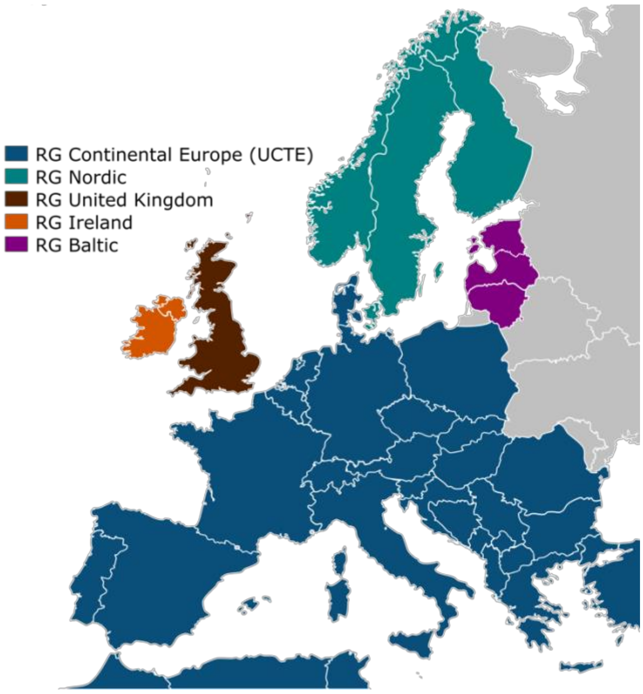


Figure 1: Map of European Transmission System Operators Organizations. From [13]. Usage authorized by Sonja Berlijn.

Chapter 3: The future of the Nordic synchronous area

3.1 The European Green Deal and electrification

In the report "An EU energy outlook for 2050" it has been estimated that the incentives, funded through the European Green Deal, will contribute to cut the CO_2 emissions by 53% within 2050 relative to the emissions in 1990 [14]. To achieve this reduction, the energy sector needs to switch over to electrical energy, both in power generation and end-user power consumption. Electricity is the preferred energy carrier because it is flexible, has high efficiency, and zero emissions at end-use. These are some of the reasons why the ENTSO-E, in their Ten-Year Network Development Plan, state that electricity will become the leading energy carrier in the future [15].

It has been estimated that the EUs power generation sector will undergo a major transformation towards decarbonization, with electricity playing a major role. 83% of the net electricity generated in 2050 will be CO_2-free, compared to 55% in 2015. 72% will be coming from renewable energy sources [14]. At the same time, the use of conventional thermal power plants will decline rapidly, acting as backup power by 2050. These numbers are supported by WindEurope's estimation of the power mix in Europe in 2050, given in Figure 2 [16]. The increased share of renewable power generation will continue to rise, with wind and solar contributing with 36% and 15% of all the power generation in Europe respectively, by 2050.

Major electrification of the end-user is also predicted to happen, leading to a 27.5% increase in electricity demand by 2050, compared to 2015. 60% of this increased demand comes from the transport sector. Electrification of the industry sector will also be a contributor to the increase in electricity demand [14].

Electrification will be an important part of Europe's goal towards decarbonization. A reliable and stable European transmission grid is therefore vital to achieve total decarbonization of all energy sectors.

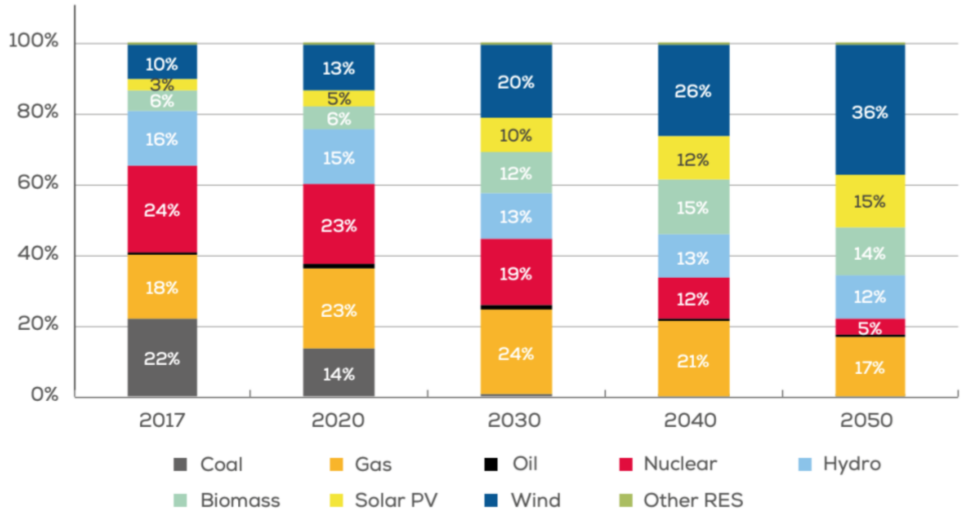


Figure 2: Bar plot showing the energy mix in Europe's electricity generation from 2017 to 2050. Research and estimates were done by DNV GL and WindEurope. From [16]. Usage authorized by Ivan Komusanac.

3.2 Challenges for the Nordic synchronous area

EU's plans to become carbon-neutral have a big impact on the future of the Nordic power system. These changes introduce challenges for the Nordic TSOs.

In the Nordic – as in the rest of Europe – the share of renewable energy is expected to increase, with installed wind power expected to triple in the 2010 to 2025 period [17]. Concurrently, thermal power plants in the Nordic, such as coal and gas, are expected to be decommissioned [18]. Replacing controllable thermal power plants with uncontrollable, intermittent renewable energy results in unreliable power production, which is a major challenge for the TSOs.

Challenges at the consumer end of the power grid will also become apparent. Incentives to decarbonize the transport sector have increased the share of electric vehicles sold in Norway. In 2020, 54.3% of the vehicles sold in Norway were electric [19]. Electrification of the industrial sector has also been pointed out as a possibility to cut CO_2 emissions in Norway. Implementing the measures presented in a report from the Norwegian Water Resources and Energy Directorate (NVE) would reduce the emissions from the industry sector by 18%. This implementation would, however, also increase the current power consumption by 10% [20]. These new and bigger loads in the electrical system could result in high consumption in a short period of time putting strain on the grid. Some of these loads inhabit a varied consumption pattern, increasing the difficulty of maintaining a stable frequency.

Another challenge is the increased number of HVDC interconnectors between the Nordic synchronous area and other countries. The number of interconnectors is expected to increase by 50% within 2025 [17]. More interconnectors increase the complexity of the power system and may introduce larger and more frequent intra-hour imbalances through HVDC ramping. This will result in more forecasting errors [17].

Weather conditions have been found to cause problems for grid operators [21]. As more extreme weather and more varied weather patterns are expected to occur due to climate change [2], this will further increase weather-related problems. High wind during storms, lightning strikes, cold waves, and heat waves are all conditions that can cause accidents on the grid, increasing the chances of faults and disturbances and resulting in more unforeseen outages.

3.2.1 Challenges for the Nordic Power System

One of the main challenges, leading up to 2025, outlined in "Challenges and Opportunities for the Nordic Power System", is frequency quality [17]. Frequency quality is an indicator of system security. If the frequency quality is high, the system will be able to maintain a stable operation during imbalances and disturbances. Repeatedly, large deviations in frequency, however, indicate low system security. This increases the risk of triggering the system's automated load and production shedding, which is undesired.

Figure 3 displays how the frequency deviation in the Nordic synchronous area has evolved since 2001. It shows that the number of minutes where the frequency of the system is outside the normal operating band has increased over the years. This trend of low-frequency quality is expected to increase leading up to 2025 [17]. The reason for this is the increased proportion of intermittent renewable production in the generation portfolio, more varied consumption patterns, and HVDC ramping.

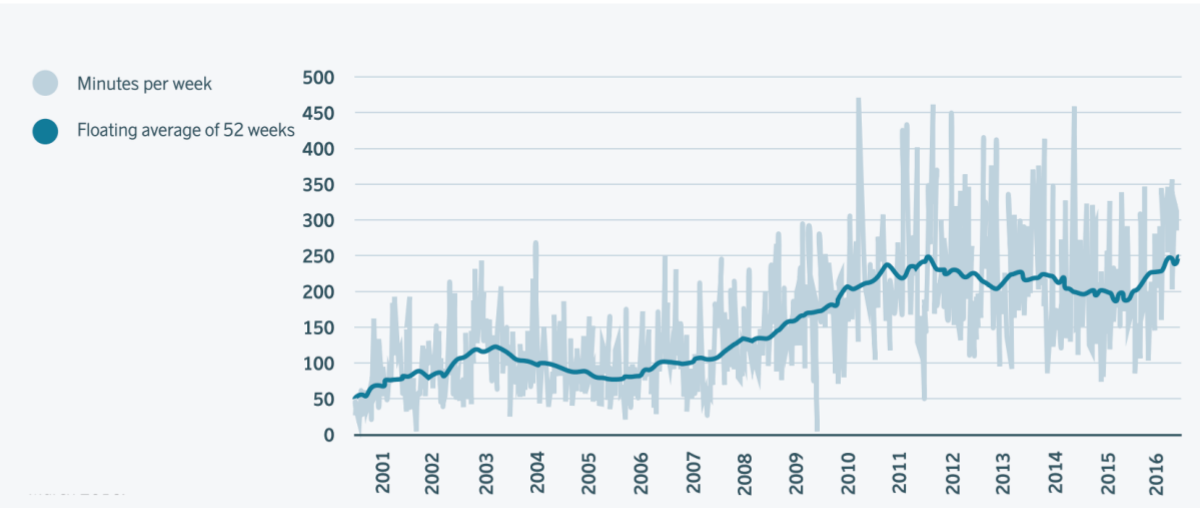


Figure 3: A graph of the grid frequency in the Nordic synchronous area, in the period from 2001 to 2016. It displays the number of minutes per week where the frequency is outside the normal frequency band (49.9–50.1 Hz). The y-axis displays the number of minutes, and the x-axis displays the year. From [17].

Problems will arise both in the generation, transmission, and consumption of electrical energy. More irregular and less dependable electricity generation, higher rate of faults on the grid due to extreme weather, more interconnected power system with increased power exchange with other countries, and new and larger loads with varied consumption patterns are all elements that introduce challenges for the TSOs and result in a more complex power system. These challenges will have to be addressed to ensure a reliable electrical grid in the future.

3.2.2 Solutions for the Nordic Power System

In ENTSOE-E's report, "Research, Development & Innovation Roadmap 2020 – 2030", these challenges are being addressed [22]. The challenges are divided into six flagships, where each flagship outlines milestones for TSO's and stakeholders spread over a ten-year period. Flagship 6 focuses on how to enhance control center operation and integrating information and communications technology infrastructure, to cope with a more complex and interconnected energy system. Figure 4 displays the flagship with its associated milestones. This master's thesis will concentrate on the milestones indicated with black circles.

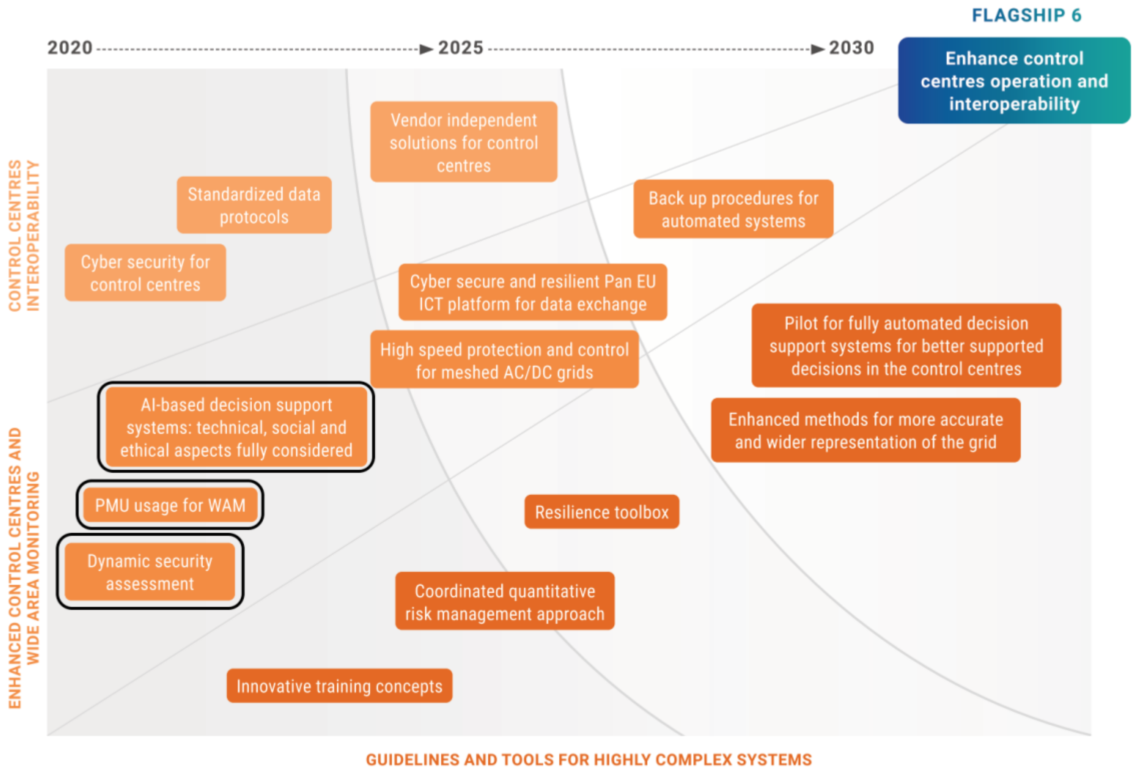


Figure 4: The milestones in ENTSO-E's Flagship 6 are presented. All the milestones are placed on a time axis, indicating when these are expected to be addressed. The milestones in focus for this master's thesis are the ones encircled by a black circle. From [22].

3.3 The N-1 criterion

Currently, the power system reliability in the Nordic synchronous grid is based on the N-1 criterion. This means that if one power line in the grid falls out, the grid will continue to deliver power. The N-1 criterion is a deterministic way of operating the grid. Since the power system is becoming more and more dynamic, the Nordic TSOs need to adopt a probability and risk-based approach towards power system reliability. Continuing to operate the grid in a deterministic way may lead to bad investment decisions, either overinvesting or investing in the wrong areas. By supplementing the current reliability measures with a probability-based approach, investing decisions can be made based on probabilities. This allows TSOs to assess the consequences of grid failures, in terms of the cost of the power interruptions to customers. Implementing such an approach could also help grid operators maintain a secure supply of electricity through probability-based decision support tools. This tendency – moving from a deterministic to a stochastic way of operating the grid – forms the basis for the Nordic TSOs next project.

3.4 The NEWEPS project

As a response to the current and upcoming challenges, the Nordic TSOs will initiate a project termed “Nordic Early Warning Early Prevention System” (NEWEPS). This will be initiated at the beginning of 2021 and will build on flagship 6 in ENTSO-E’s roadmap. With this project, the Nordic TSOs are aiming at developing a system that will improve stability monitoring, control, and visualization of the power grid. The final goal of the project is to create a prototype system that can assist grid operators with keeping the power grid at balance by providing decision support through real-time information about the grid. Certain tasks, such as voltage instability detection, power oscillation monitoring, and preventive controls, could also be automated. It is not necessarily expected that this prototype will be reliable enough to be implemented into the grid. Rather, it will serve as a system from which the TSOs will be able to learn, helping them outline the requirements for a future early warning system. Figure 5 shows a conceptual overview of the project.

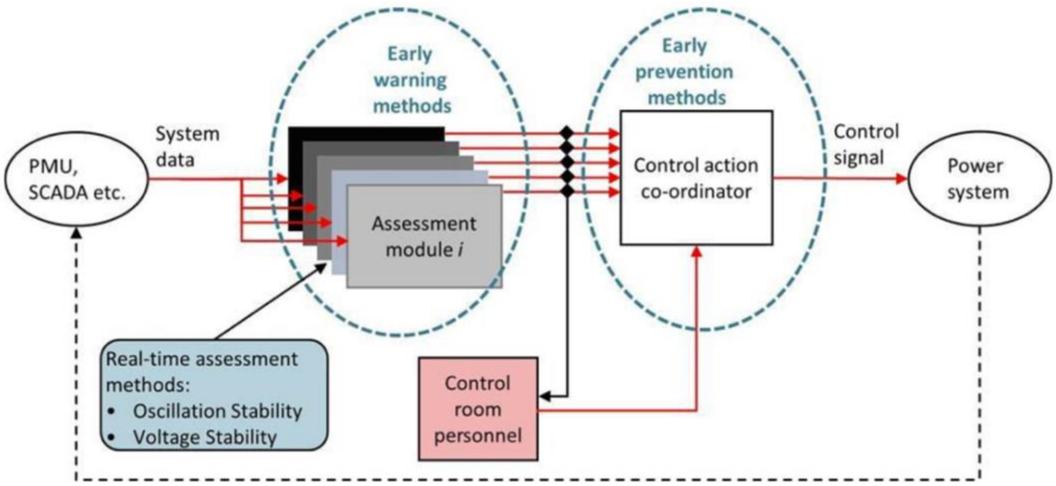


Figure 5: Conceptual overview of the NEWEPS project. From [9].

The system will use data from PMUs and Supervisory Control and Data Acquisition (SCADA) platforms as input. This data is the system state information for the assessment modules. In this project, the focus is on voltage and frequency stability, and damping of electromechanical oscillations. However, the system will be built to have a modular structure, meaning that new assessment modules coming in the future will be possible to integrate. This thesis will be a contribution to the part marked as “Early warning methods” in Figure 5.

Chapter 4: PMU theory

4.1 Phasors

Current and voltage generated from commercial alternators (or generators) can be expressed as nearly perfect sine waves:

$$x(t) = X_m\, cos(\omega t\ + \delta) \tag{2}$$

Here, X_m is the peak value of the sinusoidal signal, also known as the amplitude, ω (= 2πf) is the frequency of the signal in radians per second (angular frequency), δ is the phase angle when referenced to *cos(ωt)* and *t* is the time. When calculating active and reactive power in Alternating Current (AC) circuits, it is conventional to refer to the Root Mean Square (RMS) value, also called the effective value [23]. The RMS value is defined by

$$X_{rms} = \frac{X_m}{\sqrt{2}} \tag{3}$$

Eq. 2 can be rewritten and represented by a complex number, ***X***, known as its **phasor** representation.

$$\boldsymbol{X} = \left(\frac{X_m}{\sqrt{2}}\right) e^{j\delta} = \left(\frac{X_m}{\sqrt{2}}\right) \angle\delta = \left(\frac{X_m}{\sqrt{2}}\right) [cos\ \delta + j\ sin\ \delta] \tag{4}$$

A phasor represents the magnitude and the phase angle of a sinusoidal waveform [24]. The phase angle is the distance from the peak of the sinusoidal signal and a specified reference. The reference is a phasor at a fixed point in time. For example, in Figure 6 the reference is at t = 0. In Figure 7 the phasor representation of the waveform in Figure 6 is displayed. The magnitude of the phasor is related to the amplitude of the sinusoidal waveform. Although Eq. 4 uses the RMS values of the waveform, usage of the amplitude has also been observed in the literature.

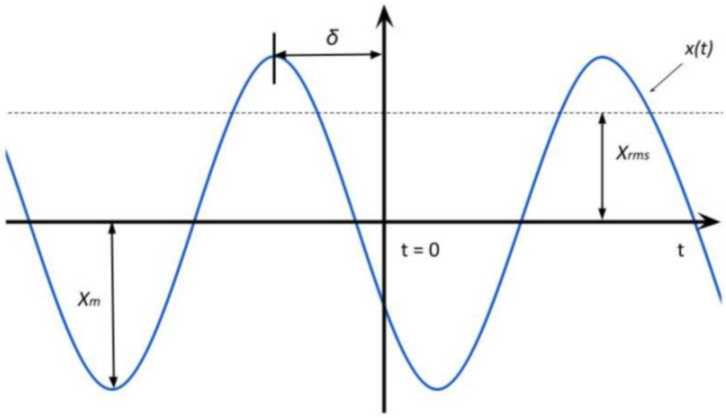


Figure 6: A graph displaying a sinusoidal waveform, *x(t)*, with a phase angle, *δ*, an amplitude X_m and an RMS value, X_{rms}. The reference point in this graph is t = 0. The y-axis represents the magnitude of the waveform. This can for example be voltage or current. Adapted from [25].

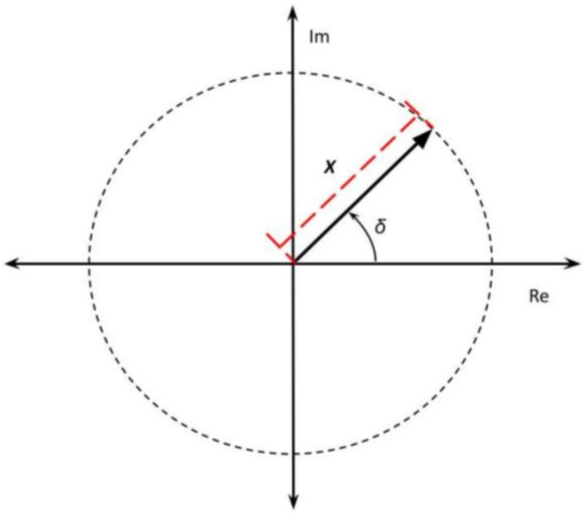


Figure 7: A graph displaying the phasor representation of the waveform. The y-axis is the imaginary axis (Im) and the x-axis is the real axis (Re). Adapted from [25].

The phase angles of the other measurements are then compared to the reference and the difference is computed, giving the relative phase angles with respect to the chosen reference. This is displayed in Figure 8 [26] [24].

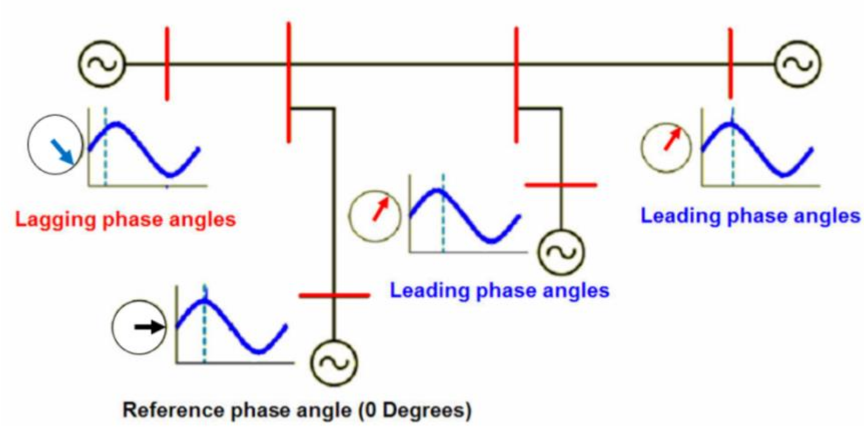


Figure 8: The figure displays a snapshot of a grid line having four PMU's. The second PMU from the left is set as the reference. From [24]. Usage authorized by Ken Martin.

To determine the relative phase angles of the different measurements, all phasors are time-stamped to an accuracy of a microsecond and synchronized through signals from global positioning system (GPS) satellites or other equivalent time sources. Time-synchronized phasors are called **synchrophasors**. Because synchrophasors are synchronized with each other, measurements taken at different locations in the grid can be time-aligned, making it possible to find the relative phase angles between many different points in the system. The sensors measuring the synchrophasors are called PMUs.

4.2 Phasor Measurement Unit - PMU

The prototype system of the NEWEPS project will, as mentioned, be using data from PMUs as input. The PMU technology has been chosen because it has been shown that it enables the use of advanced applications which can contribute to benefits in monitoring, protection, and even control of electrical power systems. This is possible because each PMU samples 30 to 60 synchronized measurements per second [5]. With the information stored in every phasor measurement, the PMUs can calculate parameters such as frequency, active power, and reactive power. Because of the low latency associated with the synchrophasors and the high frequency at which the measurements are taken, grid operators get a comprehensive near real-time view of wide areas of the grid. Abnormal system conditions, such as oscillations and voltage instabilities can therefore be identified nearly instantaneously.

4.2.1 Old SCADA system vs new SCADA system

The system used by the Nordic TSOs is a SCADA system that uses data from remote terminal units (RTU). An RTU is a monitoring device such as a PMU. The RTU, however, samples data at a much lower rate compared to a PMU. This sampling speed might occasionally be so low that transient disturbances are not detected in the power grid. Moreover, the data is asynchronous making it difficult to retrieve phase angle differences from the buses in the network [27]. The differences between the old and the new SCADA system is summarized in Table 1. Figure 9 visualizes the difference in sampling rate between these two systems.

Table 1: Comparison between RTU based SCADA and PMU based SCADA systems. From [5].

Attribute	RTU based SCADA	PMU based SCADA
Measurement	Analog	Digital
Resolution	2-4 samples per second	Up to 60 samples per second
Observability	Steady state	Dynamic/Transient
Monitoring	Local	Wide area
Phasor Angle Measurement	No	Yes

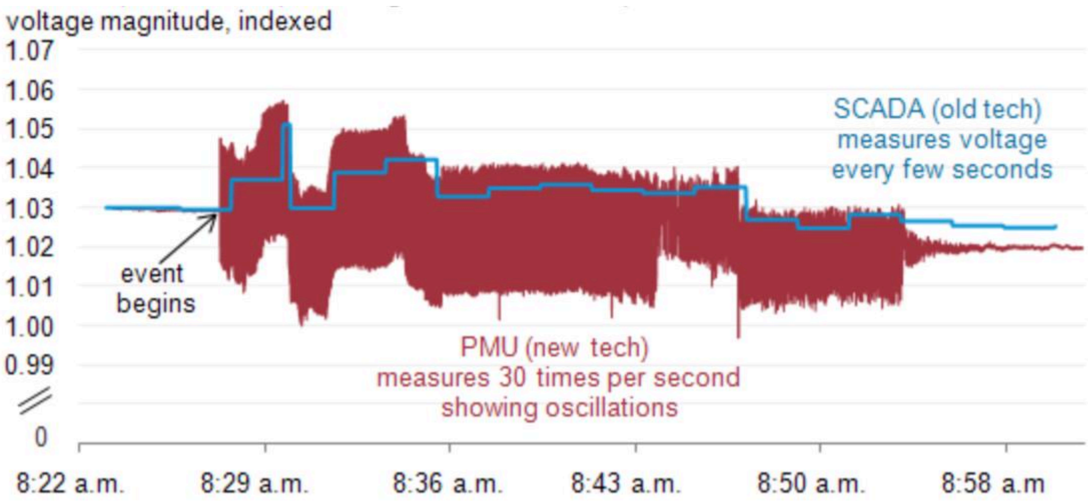


Figure 9: The figure displays the difference between a RTU based SCADA system (blue) and a PMU based SCADA system (red). From [28].

Currently, there are 265 PMUs installed in the Nordic power system. 120 of those are installed in the Norwegian grid, and an additional 60 are planned to be installed in the coming years. This shows that Statnett and the other Nordic TSOs are confident that a PMU based SCADA system will be part of the future power system [29].

Chapter 5: Research method

5.1 Literature review

The main research method used in this master's thesis is the literature review method. The literature review is divided into three parts. Firstly, it was used to gain knowledge about the theory behind ML. The books "Python Machine Learning" [30] and "Deep Learning With Python: Develop Deep Learning Models on Theano and TensorFlow Using Keras" [31] are primarily used as references for this section. To help fill in the gaps, acknowledged web resources are used.

In the second part, the same method was used to find relevant articles about the classification of faults and disturbances and system stability. The results from this literature search lay the foundation for the different categories defined later in this thesis, namely pre- and post-fault. The author was motivated to do such a categorization after performing a preliminary literature search in the databases IEEE Explore and Google Scholar about state-of-the-art ML algorithms used for predicting system stability and fault events in the power systems. This type of categorization has not been observed in the literature. Because the NEWEPS project focuses on the use of PMU data, the keyword "*pmu*" was used in every search alongside a combination of the following keywords: "*machine learning*", "*predict*", and "*power system*". This part is accompanied by information from discussions with experts in the fields of ML and power grid operation. To get an understanding of Statnett's current usage of ML technology, conversations with both a senior advisor at Statnett, who is the main contributor for the IMPALA project and a research manager at SINTEF Energy Research, who works with their Early Warn project, were carried out. A conversation with an experienced grid operator at Statnett was also performed to get information on what kind of decision support grid operators in the future would need to maintain a reliable grid.

In the third part, articles about ML models mainly using PMU data for predicting fault events and system stability in the power systems are reviewed. Through the preliminary search, it was established that there is a vast amount of literature and research done on ML models for power systems. To perform a manageable research some limitations were set:

- Articles about the detection of bad PMU data were omitted.
- Articles about cybersecurity were omitted.
- Articles about power imbalances and load and generation forecasts were omitted because the IMPALA project is concerned with these challenges.
- Articles about voltage stability were omitted. Fellow student, Krishna Solberg, has written a master's thesis about this topic in relation to the NEWEPS project [32].
- Articles about non-predictive models used for fault identification (detection and localization) were omitted. There has been written a lot about this topic, as this is a very useful decision support tool for grid operators. They are omitted from this

thesis because of time considerations. Some articles, however, present models capable of both fault localization and fault prediction. These will be included.
- A chapter about optimal PMU placement in the grid is oftentimes included in the articles. This will, however, not be discussed in this thesis.
- Articles published before 2014 are not included because the goal of this thesis is to find state-of-the-art models.

The preliminary search alongside the specification of the pre- and post-faults categories made it possible to perform a more specific literature review search. This literature search was done in Google Scholar. A combination of the following keywords was used: “pmu”, “predict”, “forecast”, “power system”, “machine learning”, “fault”, “disturbance”, and “contingency”. The following is an example of a search string used for the post-fault category:

pmu **AND** (predict **OR** forecast) **AND** "power system" **AND** "machine learning" **AND** “power system stability”

And for the pre-fault category:

pmu **AND** (predict **OR** forecast) **AND** "power system" **AND** "machine learning" **AND** (fault **OR** disturbance **OR** contingency)

To narrow the literature research even further, it was decided that only a handful of ML algorithms would be considered. The selection of these algorithms was based on the preliminary literature search and the research done in [33], an article written by researchers from Statnett, SINTEF, and NTNU. The chosen ML algorithms are:

- Linear Regression (LR)
    - Ordinary Least Squares (OLS)
- Support Vector Machine (SVM)
- Feedforward Neural Network (FFNN)
    - Multilayer Perceptron (MLP)
    - Radial Basis Function Neural Network (RBFNN)
- Decision Tree (DT) and Random Forest (RF)
- Recurrent Neural Network (RNN)
    - Long Short-Time Memory (LSTM)

Each ML model will be assessed based on its performance and response time, the methods used by the authors, and how relevant the model is for the NEWEPS project. This will be considered in a discussion section following each presented article in Chapter 8. In a summarizing discussion chapter following the literature review, each category of models will receive a Technology Readiness Level (TRL) evaluation. The TRL of the Nordic TSOs current implementation of ML models in each category will also be determined. Here the relevancy of each category for the NEWEPS project will be assessed by answering the sub-research questions.

5.2 Technology Readiness Level

TRL is a method used to decide how far a certain technology is from being implemented in real-world operations. There are several different TRL scales, but in this thesis, the TRL scale from Statnett will be used. Figure 10 and Table 9 (in the Appendix) visualize and explain the different phases in Statnett's TRL scale.

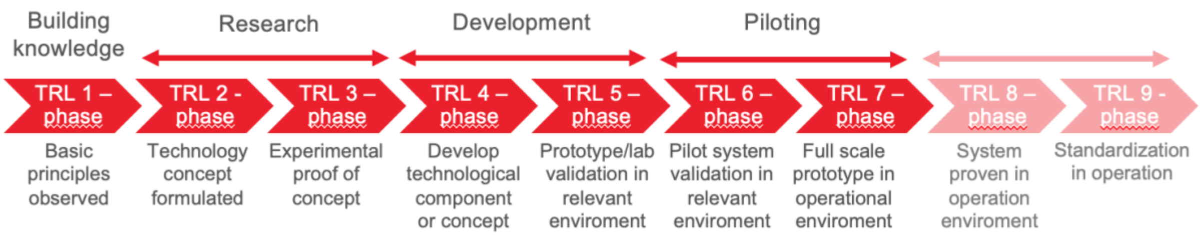


Figure 10: The figure displays the different TRLs going from 1 to 9. From Statnett.

5.3 Case study

A case study is included to exemplify some of the methods reviewed in the literature study. Because of time constraints and lack of data, implementation and testing of ML algorithms were not prioritized in this thesis. A significant amount of time was rather used to perform a simulation of a typical power grid fault. This simulation was performed in the power system software tool PSS/E from Siemens. The necessary skills to perform this simulation was found in the PSS/E Program Operation Manual [34]. The program includes a vast number of features including power flow simulation, dynamic simulation, and contingency analysis. A case from Texas A&M is also included [35]. More details on the case study are given in Chapter 9.

Chapter 6: Machine Learning theory

ML algorithms are algorithms that can automatically learn from data. By giving such an algorithm sample data, it can find patterns in the given data, producing a predictive model. This sample data is called training data. When the trained model is introduced to new data it will use pattern recognition to produce a prediction. Instead of letting humans analyze large amounts of data and manually derive rules and build a model, which can be very time-consuming and at times unachievable, one can rather help the machine develop its own model. This is especially true when the amount of data needed to analyze becomes large. Large amounts of data is not a prerequisite for an ML algorithm to perform well, but it can help uncover more patterns making the model more robust [30].

Consider the following equation,

$$y = f(x) \tag{5}$$

where y is an output variable (also called response variable), x is an input variable (also called an explanatory variable), and f is a function. The goal of an ML algorithm is to find the function, f, that outputs the best predictions for the future, y, given new examples of inputs, x. When learning, the algorithm tries to estimate the function that best maps the input variables to an output variable. This estimation will oftentimes have errors. The task in ML is to reduce these errors as much as possible. Different ML algorithms work better for different problems, and it is not always apparent which algorithm will perform best without trying. It is therefore normal to try a suite of different algorithms.

6.1 ML terminology

The following terms are commonly used in the field of ML and will also be used in this master's thesis:

- Samples (observations) are the rows in the dataset. One sample could be one flower from a specific area.
- Features (attributes) are the columns in the dataset. One feature could contain the lengths of the flower's leaves (sepal length).
- Class label (target) is the column that contains the ground truth (response variable). A response variable could be the type of flower. This is oftentimes the last column in the dataset.

Figure 11 displays these terms.

	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.8	0.8	Setosa
2	4.9	3.6	1.6	1.0	Setosa
...					
50	6.6	3.7	4.6	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.9	Virginica

Samples (observations)

Features (measurements, dimensions)

Class labels (targets, ground truth)

Figure 11: The figure displays the different terms used in the field of ML. The example dataset is a section from a commonly used flower dataset. Adapted from [30]. Usage authorized by Dr. Sebastian Raschka.

6.2 Types of ML algorithms

ML algorithms are usually categorized as either **supervised learning**, **unsupervised learning,** or **reinforcement learning**. In this thesis, only supervised learning will be considered.

In supervised learning, the model is trained on data which includes both the training examples (data inputs) and the desired output signals (class labels). This training data normally consists of 70% of the whole dataset. The remaining 30% is called the test set. When training the model, the test set is set aside. The training stage is usually done over multiple iterations. Once the model is trained, it receives the unseen data from the test set as input without the class labels and makes predictions based on this input data. These predictions can then be compared to the class labels of the test set, making it possible to assess the performance of the model. If the model performs well, it can be tested in real-time grid operation, where it will receive a stream of new data. Based on these new data, the trained model will use pattern recognition to make predictions. This process is visualized in Figure 12 [30].

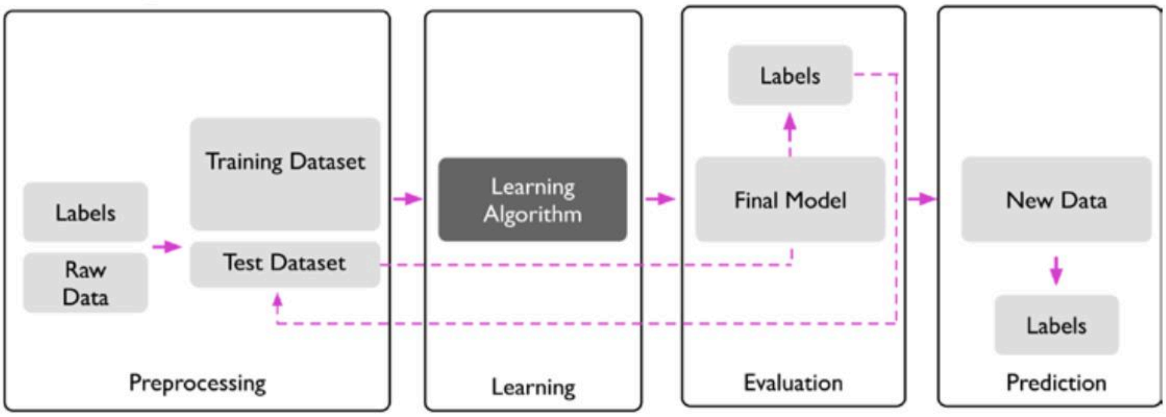


Figure 12: The figure visualizes the workflow when making a supervised ML model. From [30]. Usage authorized by Dr. Sebastian Raschka.

An example of supervised learning is email spam filtering. By training a model on a collection of emails already labeled as either spam or non-spam, it will be able to classify new emails as either one of these two categories.

In this example, the labels are discrete class labels, meaning that data inputs can be classified into two or more classes. This is called a **classification** task. If there are two classes, it is called a binary classification problem. If there are more than two, it is called a multi-class classification problem. When dealing with continuous output variables, such as an integer or a floating-point value, it is called a **regression** task. In a regression problem, the prediction from the model will be a quantity. The input data can either be real-valued variables or discrete variables. An example of a regression problem is predicting the math test scores of students, based on the amount of time spent studying [36].

6.3 Performance metrics

Performance metrics are used to evaluate how well an ML model predicts outcomes for certain problems. These metrics can be divided into metrics for classification tasks and regression tasks.

6.3.1 Classification metrics

The confusion matrix is an intuitive way of visualizing the performance of an ML model. A typical confusion matrix is displayed in Figure 13.

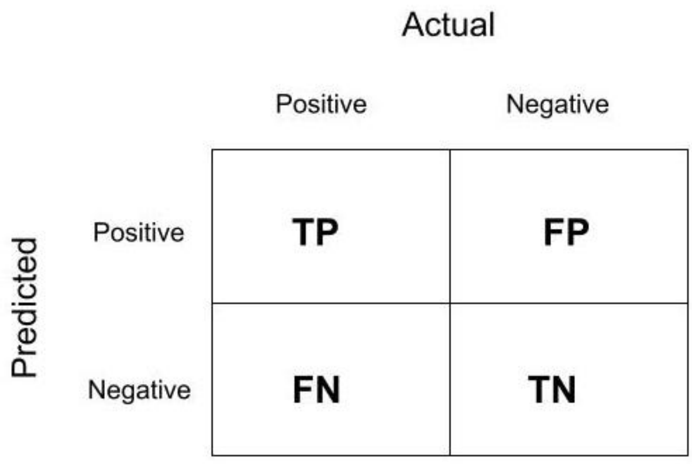


Figure 13: An example of a confusion matrix. Adapted from [37].

TP, TN, FN, and FP are short for:

- True Positive (TP): the model predicted Positive and the class label is Positive.
- True Negative (TN): the model predicted Negative and the class label is Negative.
- False Negative (FN): the model predicted Positive and the class label is Negative.
- False Positive (FP): the model predicted Negative and the class label is Positive.

The terms TP, TN, FN, and FP can be used to express performance metrics. A metric that is often used in classification problems is accuracy. Accuracy is given by

$$accuracy = \frac{TP + TN}{TP + TN + FN + FP} \tag{6}$$

Accuracy describes the number of correct predictions made over every prediction produced by the model. This metric can be used when the number of samples in each class is balanced. If this is not the case, the use of accuracy as a metric should be avoided. To exemplify this, consider the following example: given a dataset consisting of 100 samples, where 95 samples belong to class 0 and the remaining 5 to class 1, the model would get an accuracy score of 0,95 by just classifying every sample as belonging to class 0. Even though this is considered a bad predictive model, because it does not classify any sample as belonging to class 1, it achieves a high score.

A way to avoid this problem is by using the F1-score metric, which is given by:

$$\text{F1} = \frac{2 * precision * recall}{precision + recall} \tag{7}$$

Here, precision and recall are given by:

$$precision = \frac{TP}{TP + FP} \tag{8}$$

$$recall = \frac{TP}{TP + FN} \tag{9}$$

Precision is the percentage of classified Positive cases actually being Positive cases. While recall can be understood as the percentage of actual Positive cases being classified as Positive by the model. The F1-score is a value between 0 and 1, describing how precise the model is, as well as how robust it is. A high F1-score indicates a well-performing model [38].

6.3.2 Regression metrics

For regression problems, the Mean Squared Error (MSE) or the Root Mean Squared Error (RMSE) are usually used to determine the performance of the model. MSE and RMSE are defined as:

$$\mathrm{MSE} = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - y_{pred,i}\right)^2 \tag{10}$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(y_i - y_{pred,i}\right)^2} \tag{11}$$

Here *N* is the number of samples in the dataset, y_i is the ground truth of sample *i*, and $y_{pre,\, i}$ is the predicted class label for sample *i* from the ML model. MSE and RMSE describe how far off the predictions from the model are from the ground truth. A low MSE or RMSE score indicates a well-performing model [38].

6.4 Overfitting and underfitting

The goal, when training an ML algorithm, is to get a model that generalizes well. This means that the model produces sensible output when introduced to unseen input data. Understanding the terms overfitting and underfitting and how this affects the model is crucial for achieving this goal. Overfitting happens when the model learns the patterns of training data too well. Random fluctuations and outliers are picked up and are learned as concepts by the model. These concepts do, however, often not apply to new data, resulting in poor predictions. Underfitting, on the other hand, happens when the model is not able to learn any patterns in the training data. This can happen if the training data contains too few samples or that the chosen algorithm does not fit the certain problem. This would also result in predictions of poor quality. Figure 14 visualizes this.

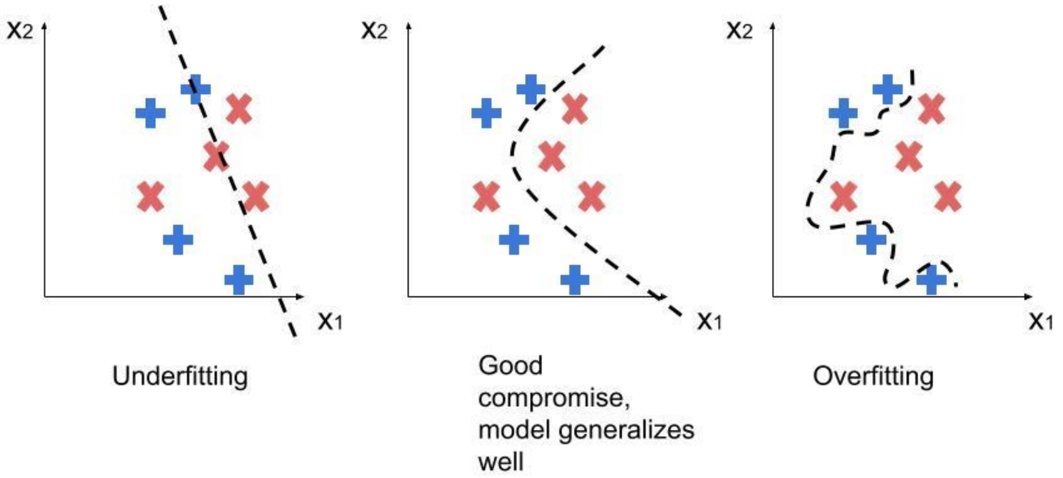


Figure 14: Three graphs visualizing the concept of overfitting and underfitting. Adapted from [30].

A common technique used to produce a more representative score for ML models is to use **k-fold cross-validation**. It is done by dividing the training data into k number of parts, training the model on k-1 parts, and testing the trained model on the last part. This is performed k number of times, computing the score for every test. The average score of the k number of test scores is then finally computed. This technique is especially beneficial for problems with small datasets [30].

6.5 Supervised ML algorithms

In the following two subchapters, examples of regression and classification algorithms will be given.

6.5.1 Supervised regression algorithm: Ordinary Least Squares

Ordinary Least Squares (OLS) is a supervised Linear Regression (LR) algorithm. Because OLS is known as a linear model, it performs very well on data that is linearly separatable. The goal of the OLS algorithm is to predict the response variable, y_{pred}, based on the explanatory variable x.

$$y_{pred} = w_0 + w_1 x \tag{12}$$

w_0 and w_1 are called the weights. These are numbers that indicate how important the accompanying feature is. If the feature has a low influence on predicting the class label its weight will become small. If, on the other hand, the feature is important for predicting the outcome correctly it will get a larger weight. These weights are getting updated for each iteration. This is how the algorithm learns. The objective of a ML algorithm is to tune these weights through a learning process such that the trained model at end can accurately describe the relationship between *x* and *y*. The model achieves this by finding the best fitting line which minimizes the residuals. In the OLS algorithm, these residuals are the MSEs between the predicted best fitting line (y_{pred}) and the data samples ((x_i, y_i)). This is displayed in Figure 15.

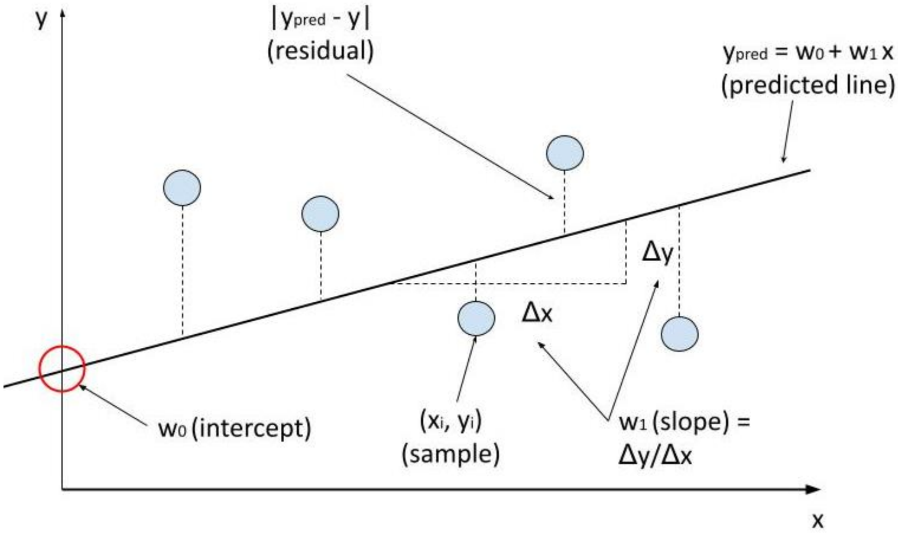


Figure 15: The figure displays how a LR algorithm works. Adapted from [30].

Eq. 13 displays a dataset with *m* number of features.

$$y_{pred} = w_0x_0 + w_1x_1 + w_2x_2 + \cdots + w_mx_m = \sum_{i=0}^{m} w_ix_i = \boldsymbol{w^T x} \qquad (13)$$

The superscript, *T*, stands for transpose. A row vector that gets transposed will become a column vector and vice versa. Thus, w^Tx will produce a number. The term w^Tx is called the net input for the model.

6.5.2 Supervised classification model: Support Vector Machine

The objective of a Support Vector Machine (SVM) is to find a hyperplane that separates the samples belonging to different classes. This is achieved by maximizing the distance from the hyperplane to the samples from each class that are closest to the hyperplane. These particular samples are called support vectors. The number of features in the dataset decides the dimension of the hyperplane. If there are two features, the hyperplane is a line, as displayed in Figure 16. Having three features produces a plane. SVMs can also be used for regression problems. These models are called Support Vector Regression (SVR) models [30].

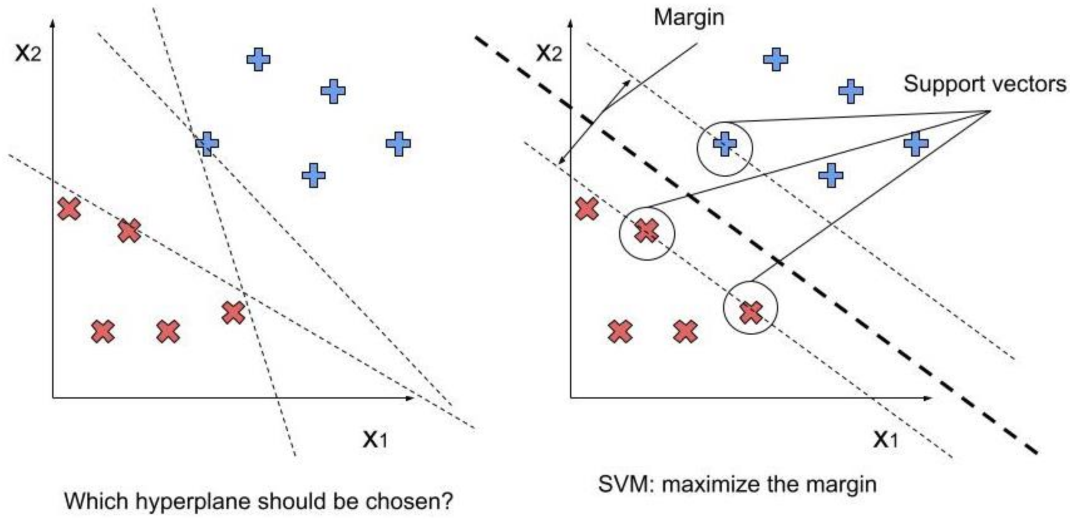


Figure 16: The figure visualizes the idea behind an SVM. Adapted from [30].

To solve problems that involve linearly inseparable data, **kernel methods** need to be used. These kernel methods create nonlinear combinations of the original features and projects these combinations on a higher dimension via a **mapping function**. Thus, the original linearly inseparable data becomes linearly separable. Figure 17 displays this. Transforming the original features onto a higher-dimensional feature space using mapping functions is however computationally very expensive. To speed up this process, different **kernel functions** are used. The most common one is the Radial Basis Function (RBF) [30]. The RBF computes the Euclidean distance between two points to determine the similarity between them. The shorter the distance, the more similar the points are to each other. This increases the likelihood of these two points belonging to the same class [39].

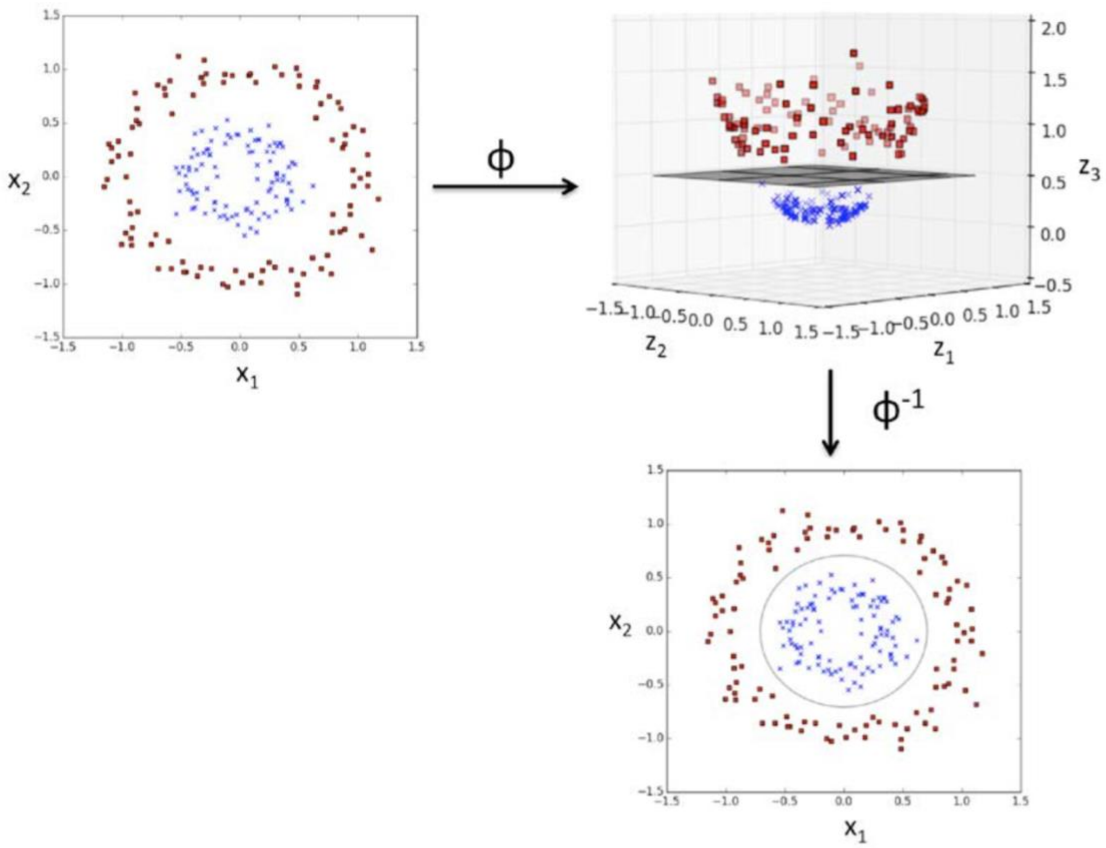


Figure 17: The idea behind using kernels to separate linearly inseparable data. From [30]. Usage authorized by Dr. Sebastian Raschka.

6.6 Decision Tree

Decision Tree (DT) is an algorithm that can be used for both regression and classification tasks. Figure 18 gives an interpretation of the idea behind how a DT works. The algorithm tries to categorize the samples in a dataset by asking a series of questions. The questions that are asked to classify each sample, are based upon the features of the dataset. Each question in the model is called an **internal node**. From each internal node, the tree splits into **branches**. When an internal node does not split anymore, it is called a **leaf node**. This happens when a certain predefined criterion is met. An example of such a criterion is the maximum depth of the model, which refers to the length of the longest path from a root node to a leaf node. An internal node will also not split anymore if all the samples in the internal node are classified as belonging to one certain class. This is called a **pure leaf** [30].

The algorithm's goal is to only have pure leaves. It reaches this goal by minimizing the impurity at each split. There are multiple impurity measures, but in this thesis, the Gini impurity, abbreviated as I_G in the following equation, will be explained

$$I_G(t) = 1 - \sum_{i=1}^{c} p(i|t)^2 \tag{14}$$

In this equation, $p(i|t)$ is the proportion of samples belonging to class i for a particular node t. c is equal to the number of classes. Thus, in a binary class setting c is equal to 2. If all the samples in one node belong to the same class, i.e., $p(i = 0|t)^2 = 0$ or $p(i = 1|t)^2 = 1$, the Gini impurity equals 0. A node having samples distributed 50:50 between the classes will have $p(i|t) = 0.5$, and thus $I_G = 0.5$. The Gini impurity tells how mixed the class labels are in each node. The algorithm starts by finding the split that is associated with the lowest Gini score. A Gini impurity of 0 results in a pure leaf [30] [40].

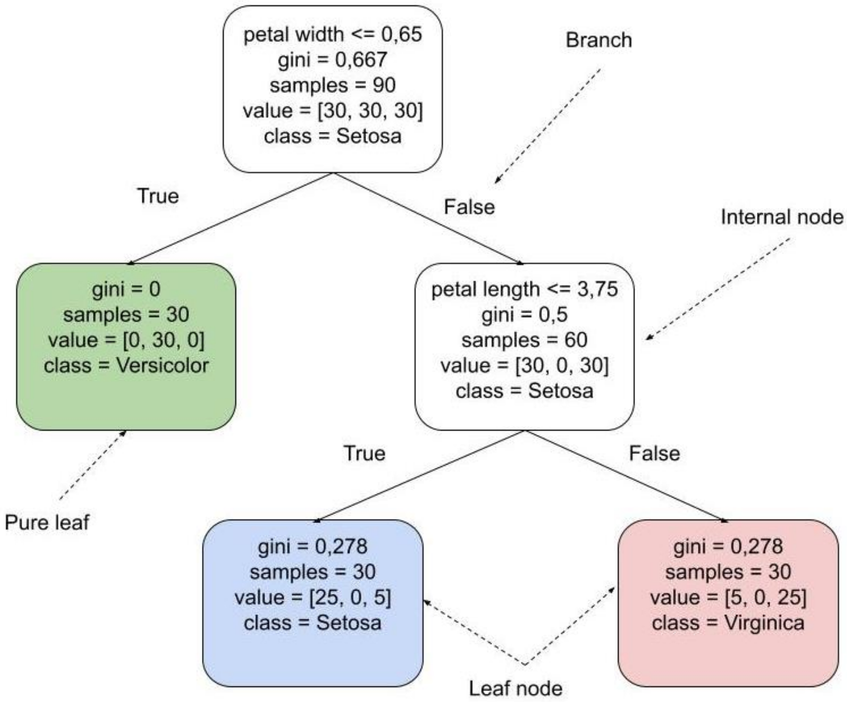


Figure 18: Figure showing the concept of a DT. Adapted from [30].

When using DTs for regression tasks the impurity measure must be changed. A common impurity measure for DT regression algorithms is the MSE.

Having too many pure leaves is an indication of an overfitted model. A common solution to this problem is to prune the tree. This means setting a maximum depth limit for the tree. Another solution is to use a Random Forest (RF) algorithm.

6.6.1 Random Forest

The RF algorithm can be thought of as an ensemble of multiple DTs. The algorithm solves the problem of overfitting, which is oftentimes associated with individual DTs, by taking the average of multiple deep DTs. This results in a model that generalizes well and is less susceptible to overfitting. A common way of taking the average of multiple DTs in a classification problem is through majority voting. For every new sample, each trained DT predicts the class that the sample belongs to. The class that gets the most predictions, or in this case, “votes”, will be chosen as the final class [30].

6.7 Deep learning

Deep learning is a subfield of ML which encompasses algorithms that are inspired by the structure of the neural networks in the human brain. These algorithms are called Artificial Neural Networks (ANN). If the connections in an ANN-based model do not form a cycle, it is called a Feedforward Neural Network (FFNN).

6.7.1 Feedforward Neural Networks

6.7.1.1 Multilayer Perceptron

The most basic form of an FFNN is the Multilayer Perceptron (MLP). A representation of such a model is given on the left side in Figure 19**Error! Reference source not found.**. The MLP, as most ANN-based models, uses artificial neurons as building blocks, displayed on the right side in Figure 19. Each neuron receives weighted inputs and computes an output using an **activation function**. The weights between the layers in a MLP are small numbers in the range 0 to 0,3 [31] randomly initiated before the network is trained. The sum of the inputs is passed through the activation function. The activation function can be thought of as a threshold that decides if the neuron should be activated or not. For example, if the summed input is larger than or equal to 0,5 the neuron will output 1. If not, it will output 0.

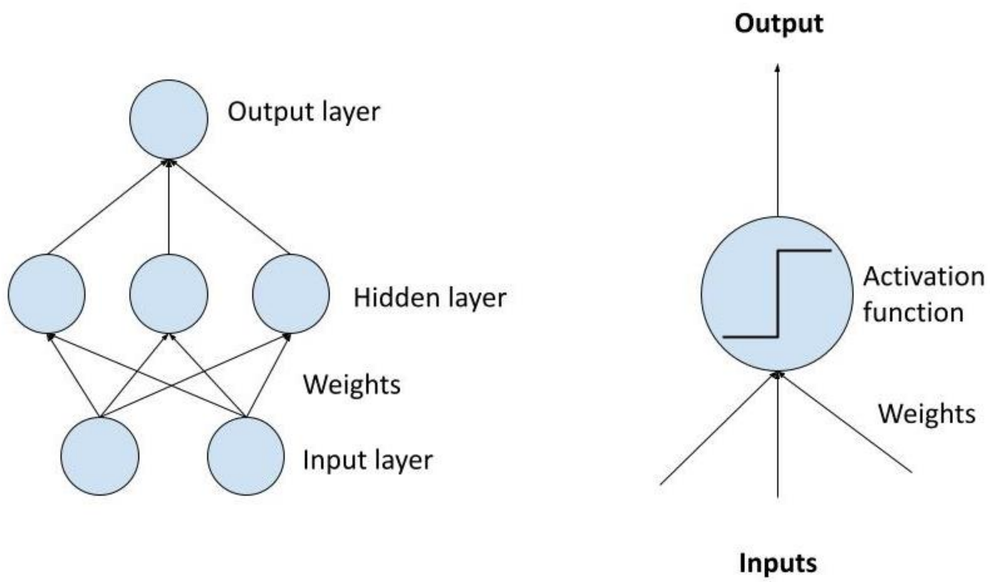


Figure 19: Left side: the structure of a simple MLP. Right side: a single neuron with an activation function. Adapted from [31].

In Figure 19 there are three layers: **input layer**, **hidden layer,** and **output layer**. The number of neurons in the input layer equals the number of features or columns in the dataset.

These neurons only pass the input value to the hidden layer and do not contain any activation function. All the preceding layers except for the final layer are called hidden layers. In the figure, there is only one hidden layer. Increasing the number of layers increases the networks' ability to learn patterns in the dataset. The downside of deeper networks is the increased requirement for computing power and the increased probability of overfitting. Increasing the number of neurons in each layer has roughly the same effect on the network as increasing the number of layers. The last layer is called the output layer. This layer is responsible for producing a value or a vector that corresponds to the certain task. Obtaining the right format for the output can be determined by modifying the number of neurons in the layer and the type of activation function [31]. The following list describes the activation functions used for the different types of ML algorithms:

- ***Regression***: Usually has one neuron in the output layer without any activation function. This results in a continuous output variable.
- ***Binary classification***: Usually has one neuron with a **sigmoid activation function** in the output layer. This results in a value between 0 and 1 which represents the probability of the model predicting the primary class. This is used in the MLP algorithm.
- ***Multiclass classification***: Usually has multiple neurons, one for each class, in the output layer. The activation function used is usually the **softmax function**, which gives the probability of class membership for each class label.

6.7.1.2 Radial Basis Function Neural Network

Another type of FFNN is the Radial Basis Function Neural Network (RBFNN). In a RBFNN, the activation functions inside the neurons in the hidden layer are RBFs. The RBFs compute the Euclidean distance from the point of the sample being evaluated to the center of each neuron. This creates clusters around the center of each neuron. Each cluster can be thought of as a class label. For example, if a data sample is inside the cluster of node 1 it will be predicted by the model to belong to class 1 [41].

6.7.2 Recurrent Neural Network

There are some problems where FFNNs do not perform very well. These problems are associated with sequential data. An example of where such data is used is in time-series forecasting, such as predicting the price of a stock over time. The reason for this is that FFNNs do not have a memory of past seen events. Therefore, the network is not able to capture the trends over a longer period. The Recurrent Neural Network (RNN) solves this by adding recurrent loops to the architecture. In each layer, the signals do not only get passed forward to the next layer but are also passed sideways inside the layer. This is displayed in Figure 20 [30].

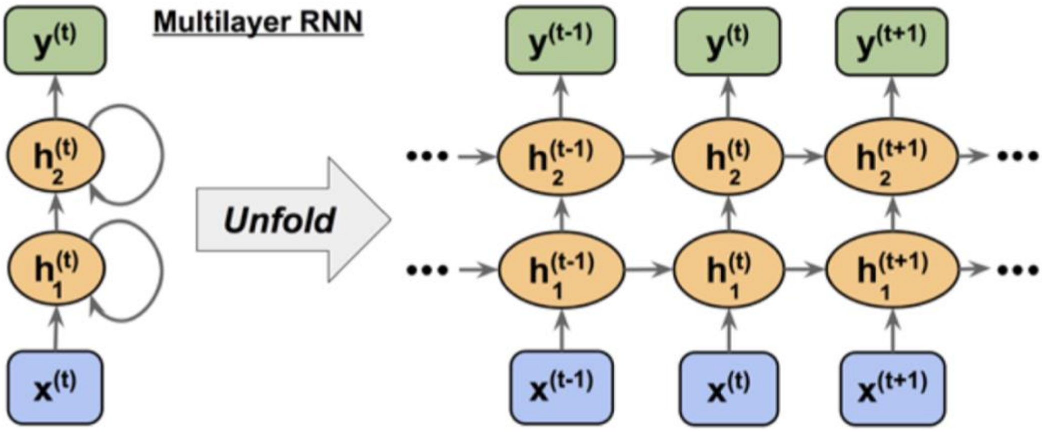


Figure 20: The figure displays the structure of a multilayer RNN. From [30]. Usage authorized by Dr. Sebastian Raschka.

In this figure, $x^{(t)}$ is the input, $h_1^{(t)}$ and $h_2^{(t)}$ are the two hidden recurrent layers and $y^{(t)}$ is the output, all at time t. As can be seen from the figure, the hidden layers do not only receive their input from $x^{(t)}$ but do also receive values from the previous timestep, $h_1^{(t-1)}$ and $h_2^{(t-1)}$. This is how the algorithm is capable of remembering past information [30].

6.7.2.1 Long Short-Term Memory

A problem that occurs when updating the weights of a deep RNN, is that the weights can become very small resulting in the network not making any progress. This is called the vanishing gradient problem. The Long Short-Term Memory (LSTM) network overcomes this problem by carrying information over many time steps, thus preventing older signals from vanishing. The building block of an LSTM is a memory cell, shown in Figure 21 [30].

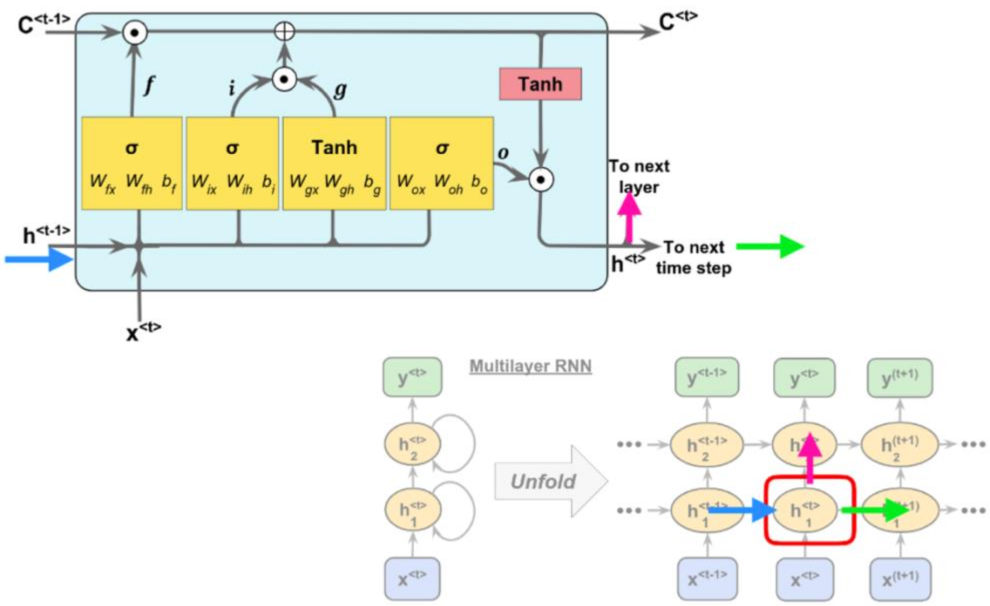


Figure 21: The figure displays a memory cell inside an LSTM network and where it is in a multilayer RNN. The arrows display how the information flows. From [30]. Usage authorized by Dr. Sebastian Raschka.

In addition to the input (x^t) and the hidden layer (h^t), the memory cell also contains the cell state (C^t) and the four separate gates (the yellow boxes in Figure 21). The cell state transports the information through all the memory cells. This can be thought of as the memory of the LSTM network. The four gates decide what information is included in the cell state. During training, these gates learn what information is important and discard the information that is not relevant. This is done through activation functions. In an LSTM network, the sigmoid (σ) and the tanh (Tanh) activation functions are used. The tanh activation function acts in the same way as the sigmoid, only that instead of squishing the input to be a value between 0 and 1, the input is squished to be a value between -1 and 1. In Figure 21, the white circle with a black dot represents an element-wise multiplication and the white circle with a cross represents an element-wise addition [30].

6.8 Times-series forecasting

The process of predicting future events based on historical data is called time-series forecasting. To transform the time-series problem into a supervised learning problem one could use the **sliding window method**. The method uses a set of time-dependent measurements as input, as seen in Figure 22. This could for example be PMU measurements. The number of measurements used as input is decided by the width of the window, i.e., the time interval. Based on the input, the method will predict a label, either a category (classification task) or a quantity (regression task) [42].

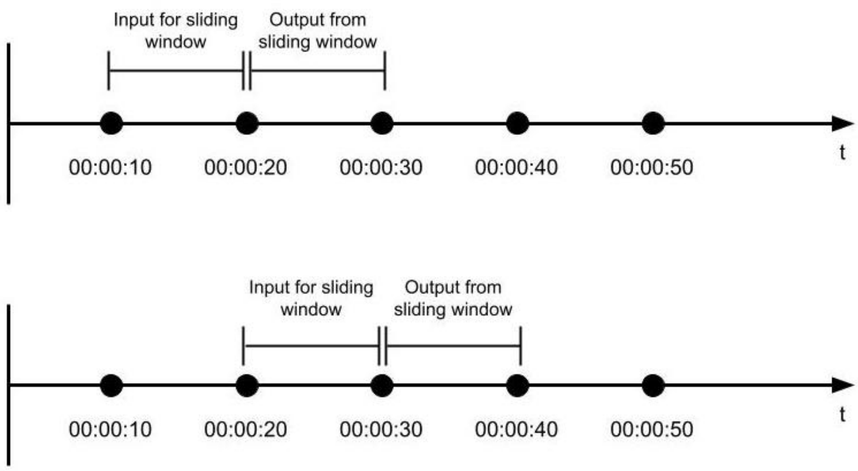


Figure 22: The figure displays how the sliding window method works. Adapted from [43].

In the literature, there is usually a distinction between static and time-adaptive sliding windows. When using a **static time window**, the author of the model uses a predefined time window that does not change. An example of this is setting a six-cycle time window. In this case, the algorithm gathers PMU data for six cycles before it computes a prediction. Here, a cycle is meant by the time it takes for the sinusoidal waveform to repeat itself (e.g., from peak to peak in Figure 6). One cycle in a 50 Hz system is therefore 0,02 seconds. When using a **time-adaptive window**, on the other hand, the algorithm increases the width of the time window as it goes. It starts with the first time-step. This could be after one cycle of PMU data. If the model produces a prediction inside the “not credible” boundary, the next time-step will be used as input. This repeats until the model either has made a credible prediction, or it meets the user-defined end time. The “not credible” boundary is also user-defined. The advantage of utilizing this method is that it may be faster than the static method [44].

6.9 Criteria for ML algorithms

There are two important parameters to consider when assessing ML algorithms. These are the **response time** and **performance** of the model. Response time is meant by the time it takes for a trained ML model to produce a prediction given an input. Performance is meant by how close the predictions from the trained ML model are to the ground truth or real-world outcome. In Table 2, timeframes for assessing different power system conditions, relevant for this thesis, are presented [45].

Table 2: The table presents the required response times for selected power system conditions. Values from [45].

Power system condition	Required response time for predictive model
Transient stability	150 milliseconds – 1 second
Dynamic (electromechanical) stability/LFOs	1 – 5 seconds
Fault/disturbance prediction	Unknown; depends on fault/disturbance

Chapter 7: Classification of faults

To categorize ML models, the different categories need to be clearly defined. The timespan of a fault can be divided into three parts, as displayed in Figure 23.

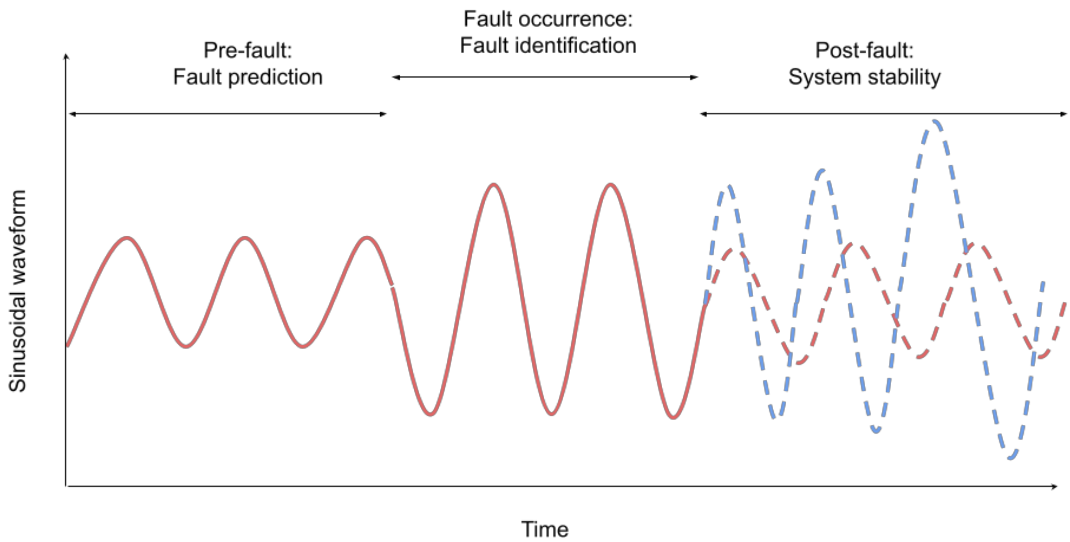


Figure 23: The figure displays the waveform for the three timeframes of a fault occurring in the power grid. The blue dotted line visualizes a case where the predictive model predicts that the system will be unstable following a cleared fault, while the red dotted line visualizes a case where the predictive model predicts that the system will return to a stable state immediately after the fault is cleared. Inspired by [45].

7.1 Post-fault/Power system stability

In this thesis, post-fault models are defined as models that predict the real-time stability of the power system right after a fault has happened. Power system stability is associated with the power system being able to regain a state of operating equilibrium after a fault. It can be divided into **rotor angle stability**, **frequency stability,** and **voltage stability** [46], as displayed in Figure 24. In this thesis, only articles concerning rotor angle stability will be evaluated. The reason for focusing particularly on this subcategory is because this category of system stability is one of the desired functionalities of the NEWEPS prototype.

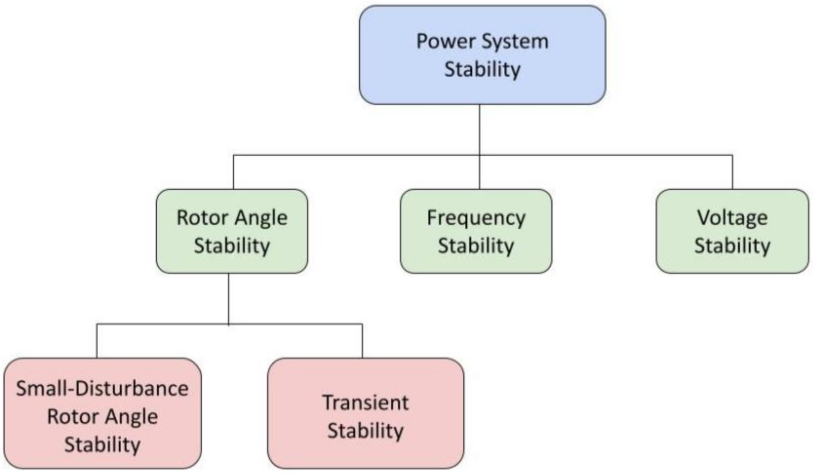


Figure 24: The figure displays the main categories and sub-categories of power system stability. Some sub-categories from the original source are omitted because it is outside of the scope of this thesis. Adapted from [46].

Rotor angle stability is associated with the damping of electromechanical oscillations inherent in power systems. The rotor angle (also called the load angle) is the angle between the axis of the stator and the rotor in the synchronous generator. It can be used to determine the active power from the generator. Larger rotor angles result in larger power output and vice versa. If the load angle is increased beyond 90 degrees, the generator will lose synchronism. The rotor angle also determines the load sharing between multiple synchronous generators connected in parallel. When in an equilibrium state, the generators are running at equal speeds. Once this balance is disturbed, oscillations in the rotor angles will appear. If these are not properly damped, the generators can fall out of synchronism and instability occurs. In a power system with sufficient rotor angle stability, the synchronous machines will therefore be able to regain synchronism after a fault. The subcategory can be further divided into large and small disturbances [46].

Large-disturbance rotor angle stability is usually called **transient stability**. A large disturbance could, for example, be a short-circuit fault in the transmission line, which results in large excursions of generator rotor angles. Assessing rotor angle excursions of synchronous machines following such a disturbance gives an indication of the transient stability of the power system. **Small-disturbance rotor angle stability** is concerned with the study of non-oscillatory instabilities and the damping of **low-frequency oscillations** (LFOs). These occur when the power system is subjected to small disturbances like small load changes and small generators tripping. In this thesis, only LFOs will be considered. LFO appears in poorly damped power systems, meaning the system is lacking enough damping torque [47].

7.2 Fault occurrence

In this timeframe, the goal is to quickly determine where the fault has occurred (fault localization) and what type of fault it is (fault classification) [8]. These models will, as mentioned before, not be covered in this thesis as these are not predictive models. However, good classification models are of great importance for both real-time grid operations and for creating real-world training data for predictive models. This category of models has been extensively researched since the late 1990s and is currently being used by multiple TSOs worldwide [45]. In the report from CIGRE [45] it was, however, not specified if these models made use of ML technology. This large difference in the amount of research and literature between fault identification and fault prediction implies that the former is a less complicated problem to solve than the latter.

7.3 Pre-fault

In this thesis, pre-fault models are models that predict upcoming faults. This could be day-ahead, intra-day, intra-hour, etc. Through the literature study, it was found that very few articles are using PMU data as input for ML models to predict upcoming faults. For that reason, some articles using non-PMU data have been included in the study. The pre-fault category is therefore divided into the following two categories: **fault prediction models** and **fault prediction models using non-PMU data**.

To be able to predict an upcoming fault, the fault needs to have a signature in advance of the fault event. It has been found that many of the most frequent faults happening in the power system have a signature indicating an upcoming fault [8] [35]. These signatures can be detected on the waveforms from the phase currents and phase voltages going in the transmission lines. The anomalies in the waveforms may be so subtle that they are easily overlooked by grid operators. Cases from a Texas A&M's Distribution Fault Anticipator (DFA) [35] project show that such anomalies occurred ranging from three days to a month ahead of the event of the major fault. In one of the cases, the final fault resulted in an hour-long blackout. Subtle signals from intermittent minor faults were produced one month ahead of this blackout event. High-fidelity recording equipment in combination with pattern recognizing ML models could be able to detect these signatures and give a warning to the grid operator, avoiding such blackouts. Some faults have more distinct signatures than others. Component failures due to humidity, aging, etc. have been recognized as faults having such signatures. An example of this is the mechanical degradation of transformers [48].

7.4 Common faults and disturbances in the Norwegian power system

Every year, Statnett releases a report presenting all the disturbances and faults that happened during this year [49]. In 2018, there were 740 disturbances in the 33 kV-420 kV grids, displayed in Figure 25. Of these disturbances, approximately 62% were categorized as either being caused by surroundings or technical equipment. In their report, Statnett has a measure on the amount of energy that would have been delivered to the customers if there would not have been a disturbance in the grid. This is called Undelivered Energy (UE). In 2018, disturbances caused by surroundings accounted for 60% of the total UE. It has been estimated that UE disturbances yearly account for 800 million NOK. This number represents the socio-economic expenses for the end-user due to the disturbances [50].

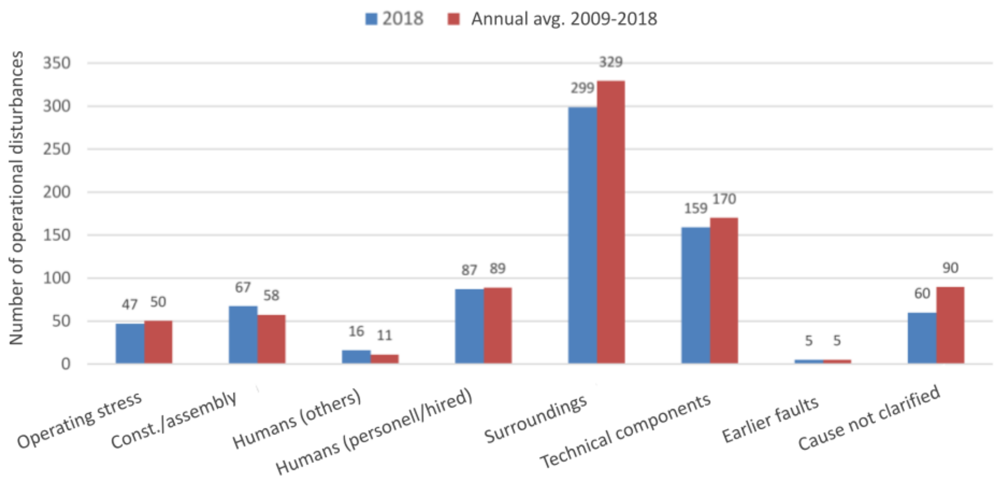


Figure 25: A bar plot displaying the number of disturbances in the transmission grid in 2018 (blue) and the annual average based on data from 2009 to 2018 (red). The disturbances are divided into the different factors that caused the disturbance (x-axis). Adapted from [49].

A disturbance can consist of one or more faults. 818 faults caused the 740 disturbances in 2018. Disturbances in the transmission line were most frequently observed, accounting for 36,3% [49]. There are several different types of transmission line faults. Figure 26 displays these types of faults. The most common ones are unsymmetrical faults. From these, the single line-to-ground fault is the most common one [51].

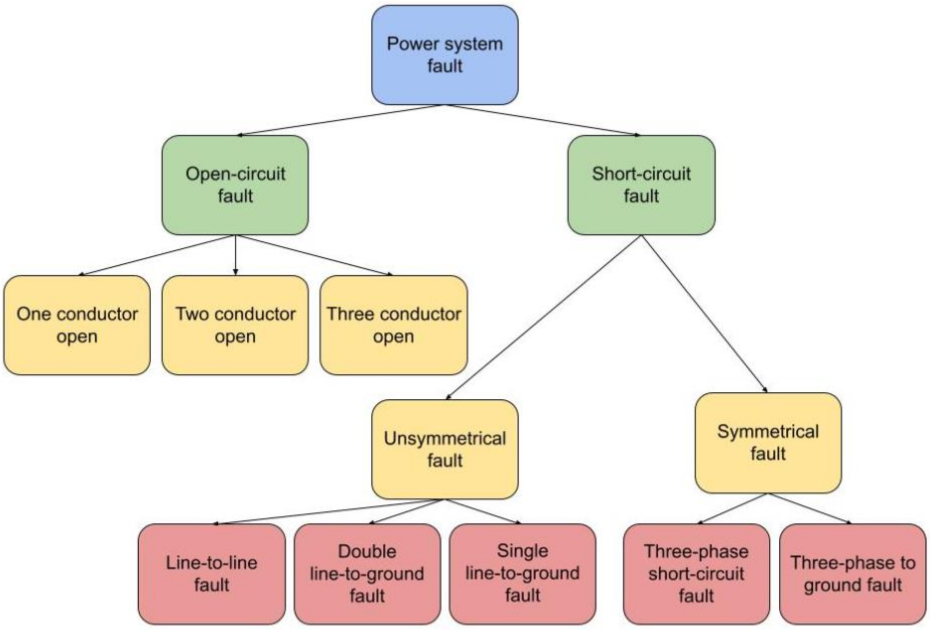


Figure 26: The figure displays the different types of faults that can happen in a power system. Adapted from [52].

From Table 3 it can be seen that vegetation and wind are two of the three primary triggering causes for disturbances caused by surroundings. Both vegetation, such as trees and twigs, and strong wind do oftentimes cause earth- and short-circuit-faults [8]. These two categories of triggering causes for disturbances have been found to produce signatures ahead of the fault [35]. Such faults are not only very costly for the TSOs but can also potentially pose great danger for the environment in which the disturbance happens.

Table 3: The table displays the triggering causes for the disturbances categorized as surroundings. Thunderstorms (and lightning strikes), vegetation, and wind are the three most common causes of disturbances. Adapted from [49].

	Number of operational disruptions				**UE because of operational disruptions**			
	Quantity		Share		MWh		Share	
Triggering causes: Surroundings	2018	Annual avg. 2009-2018	2018	Annual avg. 2009-2018	2018	Annual avg. 2009-2018	2018	Annual avg. 2009-2018
Bird/animal	13	8	4,3 %	2,4 %	32	11	4,0 %	0,3 %
Salt/contamination	0	8	0,0 %	2,6 %	0	21	0,0 %	0,6 %
Snow/ice	34	37	11,4 %	11,3 %	51	172	6,3 %	5,0 %
Thunderstorms	85	125	28,4 %	37,9 %	15	211	1,9 %	6,1 %
Vegetation	85	49	28,4 %	14,9 %	630	550	78,7 %	16,0 %
Wind	52	72	17,4 %	22,0 %	27	2 335	3,3 %	68,0 %
Other	26	22	8,7 %	6,5 %	44	108	5,5 %	3,1 %
Cause not clarified	4	8	1,3 %	2,6 %	2	26	0,3 %	0,7 %
Sum	**299**	**329**	**100 %**	**100 %**	**800**	**3 433**	**100 %**	**100 %**

The third primary triggering cause for disturbances is thunderstorms and lightning strikes. This category of surroundings does not produce a signature in the waveforms. But since this category accounts for a large portion of the disturbances, it should not be overlooked when building a predictive model. A common way to include the effect thunderstorms and lightning strikes have on the power grid, is to integrate weather forecasts in the predictive model. Articles describing predictive ML models based on weather forecasts will therefore be included in this thesis.

7.5 Previous research from Statnett and SINTEF Energy Research

There has been done some previous research in the field of predictive ML models by Statnett and SINTEF, one of Europe's largest independent research organizations.

7.5.1 IMPALA

IMPALA, a project initiated by Statnett, NTNU, and Optimeering, which launched in 2017, aimed at developing an ML system capable of predicting power imbalances in the power system. This means that rather than trying to predict faults and disturbances they wanted to predict the consumption pattern of the end-user [53][54]. Several different data inputs are used such as estimations on future production and consumption, weather data, and historical data. According to a representative from the IMPALA project, the offline testing of the system was a success and their team is currently aiming for a 15-minute window ahead prediction [55].

Their goal is to create a system that should be integrated into real-time operation. The system should be able to not only make predictions about power imbalances but also be capable of ordering the right amount of balancing power needed from a new Nordic balancing model [56], thus keeping the frequency in the power grid stable, making the grid more robust. This system will be completed by 2022. Through the IMPALA project, Statnett has gained a lot of experience in the field of predictive ML models concerning load forecasting and power imbalances. This is the reason for excluding these particular models from the literature review performed in this thesis.

7.5.2 EarlyWarn

The EarlyWarn project, a collaboration between SINTEF Energy Research, Statnett, and NTNU, which launched in 2017, aims at applying ML techniques to predict faults and disturbances in the power system [57]. The stakeholders are hoping that the use of ML techniques in a proactive fault detection system, may downsize or even prevent the adverse impacts of a disturbance. In the project, both data from PMUs and Power Quality Analyzers (PQAs) are tested. According to a representative from the EarlyWarn project [58], the team working on these models has experienced that locating and classifying faults are more easily done than predicting upcoming faults. This confirms the assumption made earlier in this thesis that producing ML models for fault prediction is more difficult than producing ML models for fault identification.

Chapter 8: Reviewed literature

The results from the literature review are divided into the defined categories, namely **post-fault models** and **pre-fault models**. These categories are further divided into articles concerning transient stability and LFO, and fault prediction and fault prediction using non-PMU data, respectively. For each subcategory, a table containing the articles found from the literature search is presented. The table contains which algorithm the author(s) used, the methods used for training and testing the algorithm, and the performance of each ML model. In each subcategory, a more in-depth look at two articles from the table is presented. This is to display the different procedures that are commonly used by authors working with these predictive tasks. The reasons for choosing these articles are given below.

Each subcategory contains a discussion section, where the performance of the ML models is discussed considering the methods used by the author. The test times and the test system used will also be considered. In this section, the data in the training and test sets and the size of the whole dataset are also examined, as these are important factors that could influence the performance. Every article from the tables will be included in the discussion.

The in-depth articles were chosen based on multiple criteria:

- The model should be state-of-the-art. The most recent articles are therefore chosen.
- The relevancy for the Nordic power system. For example, articles that use test systems where wind farms are incorporated are preferably chosen.
- Norwegian weather conditions. Articles that consider high wind are preferably chosen, as this is a common weather condition in Norway, see Table 3.
- Articles considering earth/ground faults are preferably chosen. These faults are, according to an experienced grid operator [59], very common and need to be addressed.

8.1 Post-fault models

8.1.1 Transient stability

8.1.1.1 "Real-Time Monitoring of Post-Fault Scenario for Determining Generator Coherency and Transient Stability through ANN"

In [60], the authors proposed a RBFNN to determine real-time transient stability status and identification of coherent generator groups. Identifying the coherency among generators after large disturbances can be beneficial for initiating control islanding, according to the authors. By splitting the system into smaller networks, the adverse effects of a disturbance can be minimized, avoiding a total system failure. Real-time information about generator coherency is therefore vital for the grid operators in order to initiate the proper control islanding strategies. In the article, an IEEE 39-bus test system was used for generating the training data and for testing the trained model.

For each sample, there were used six consecutive cycles of post-disturbance data from PMUs. The inputs for the model were rotor angles and voltage magnitudes. Faults were imposed at the generators, at the loads, and in the midpoint of the transmission lines, producing a labeled dataset containing 369 stable and 305 unstable cases. The faults consisted of three-phase faults with varied load patterns. Fault clearing time was set randomly between five and twelve cycles. Fault clearing time is here meant by the time used by the operator to clear the fault. The model was trained on 505 samples of the total 674 samples. Once trained, the model predicted the future rotor angles for the generators in the system and determined both the system's transient stability and the coherent groups of the generators 10^{-4} seconds after the fault was cleared.

8.1.1.2 *"A new method of decision tree-based transient stability assessment using hybrid simulation for real-time PMU measurements"*

In [61], the authors used DT and RF models to classify transient stability as stable or unstable. The major contribution in this article, according to the authors, is that they highlight the problem of misleading training and test data for models predicting transient stability. They argue that that the conventional method of generating training data for ML models, originating from phasor-based simulators, produces unrealistic data and does not catch the dynamic aspects that measuring units such as PMUs inhabit. Training algorithms on this type of data will produce models that are not suitable for practical power systems. Therefore, a hybrid simulation method is proposed. The methods consist of both a phasors-based program and an electromagnetic transient simulator, which is based on three-phase waveforms of currents and voltages.

Another problem highlighted by the authors is that the phasor estimation algorithms might produce erroneous measurements during a limited period after a disturbance. This happens under off-nominal frequency conditions. The impairment in the measurements appears as transient ripples. Samples from this limited period, called the settling time, need to be removed according to the authors. The authors display the importance of taking both of these problems into account. This was done by training a DT and RF model using the conventional method and the proposed method and then comparing the performance of these two.

An IEEE 68-bus test system and a WSCC 127-bus test system were used to generate datasets. By introducing three-phase-to-ground faults on all buses and all lines, having the fault appear at 25%, 50%, and 75% of the length of the line, 30 000 samples for each system for training and testing were produced. Fault clearing time was randomly set between four and eight cycles. The ratio of unstable to stable instances in the dataset was 1:2. The ML models used voltage magnitude as an input feature and used a 0,1 second time window. Five-fold cross-validation was used to tune the parameters of the classifiers.

Table 4: The table summarizes the methods used and the performances of the ML models associated with transient stability problems.

Article	Year	Algorithm	Fault applied	Method	Task	Performance
[60]	2017	RBFNN	Load variations and three-phase faults at three different locations	Static sliding window. Size = 0,1 s	Predict transient stability: stable or unstable	Acc = 98,36 %
					Estimate rotor angle	Avg. Error = 5,238 %
					Predict coherent generators in groups	All correctly classified
[61]	2020	RF and DT	Load variations and three-phase-to-ground faults on buses and lines	Static sliding window. Size = 0,1 s	Predict transient stability: stable or unstable	RF: Acc = 95,3-98,5 %
						DT: Acc = 92,9-96.4 %
[62]	2020	MLP, DT and Naïve Bayes classifier	Three-phase faults on buses	Time-adaptive sliding window. Avg. size = 0,08 s	Detect faults and predict transient stability	MLP: Acc = 96,56-98,02%
[63]	2018	LSTM	Load variations and three-phase faults at multiple different locations, during (N-1) contingencies	Time-adaptive sliding window. Avg. size = 0,034 s	Predict transient stability: stable or unstable	Acc = 99,98-100%

8.1.2 Discussion about transient stability

The RBFNN used in [60] gives predictions very fast, needing only six cycles of post-disturbance data and having a test time of around 10^{-4} seconds. The test time is the time that the trained model needs to give predictions using the test samples as input. The DT and RF models in article [61] used the same time window. In a 60 Hz system, six cycles of post-disturbance data result in 0,1 seconds. But in this article, the test time was not mentioned. From Table 2 it is clear that a short test time is crucial for models assessing transient stability. Both articles generated balanced datasets, which make the accuracy scores trustworthy. By changing the fault clearing time, which both articles did, the models get more robust.

The authors in [61] used 30 000 samples for training and testing, while the authors in [60] only used 674 cases. Here, one case is assumed to consist of many samples (or time steps) resulting in a larger dataset. Even though there is no definite answer on how big a dataset needs to be, using only 674 samples for such a complex problem would be deemed as being too few. In [61] a larger test system was also used compared to the test system in [60]. This increases the significance of the results in [61] as the predictive model had to process a more complex system.

From Table 4 it can be observed that the models produced great performance results. The table also displays the performance of ML models from two other articles [62] [63]. The main difference between these two types of articles is that the latter type of articles used a time-adaptive sliding window instead of a static sliding window. Although this seems to increase the accuracy score of the models, it also increases the complexity of the models, which might increase the test time. The authors did, however, not mention the test times. This tradeoff might be worth it, given the increase in performance, as this could be good enough to implement the models in real-time operation.

None of the models were tested on real-world data or used in a real-time grid operation scenario. They depend on artificially generated data using different simulation tools. This creates a stream of data, imitating real-time operation. Implementing the models in real-world grid operation would probably lower the performance. The reason for this is that most power system simulation tools do not have the capability to capture every stochastic aspect and complexities of a large, real-world power system. However, the models in [61], being trained on a hybrid simulation method, might perform better in such a situation.

8.1.3 Low-frequency oscillations

8.1.3.1 "Prediction of Electromechanical Oscillatory Parameters in Power Systems Using ANN"

In [64], the authors proposed an FFNN to predict power oscillation parameters, i.e. instantaneous frequency, instantaneous amplitude, and damping ratio. They argue that by providing grid operators with real-time information about these parameters, the amount of cascaded tripping and blackouts could be reduced. A Hilbert transform is used to compute the dataset used for training and testing. The Hilbert transform uses voltage signals from PMUs to compute the power oscillation parameters, thus creating a labeled dataset for training the FFNN. Once the model was trained on the dataset, a new separate set of samples was made (test set) to determine the performance of the model.

For simulation, a two-area 11-bus test system was used. In the system, two similar areas are connected by a weak tie (transmission line). Two scenarios were created. In the first scenario, a three-phase fault was applied to the weak tie and in the second, a load, close the weak tie, was removed. This resulted in a dataset containing 1200 samples. The test time is in the order of 10^{-4} seconds.

8.1.3.2 "Wide-area PMU-ANN-based monitoring of low frequency oscillations in a wind integrated power system"

In [65], the authors used an FFNN to predict the LFO modes and classify their localness in a modified IEEE 39-bus test system under several N-1 contingencies. The system is modified by placing two wind farms in the system. Together these windfarms accounted for 25% of the total production, affecting the damping of LFOs in the system. The output parameters from the trained model were frequency, mode index, and damping ratio. Mode index is used to rank the LFOs according to their localness. Damping ratio is a value that tells the operator if the system is classified as critical and needs to be damped to maintain the system's oscillatory stability. The authors argue that with this information, grid operators will be able to perform proper control actions within the critical timeframe.

Voltage magnitudes and angles from PMUs were used as inputs for the model. A Principal Component Analysis (PCA) was implemented, reducing the number of features in the generated dataset. A PCA retains most of the information in a dataset while reducing the training and testing times of the model. The dataset was generated through modal analysis. Modal analysis is an offline analysis tool used to characterize the dynamic evolution of a power system. This type of analyzing method is, according to the authors, time-consuming and does not perform well on larger systems.

The dataset contained 2500 samples of the first four-cycle data of voltage magnitudes and angles following the disturbance. The trained model predicted the frequency and the damping ratio of LFOs and determined if oscillations were local or inter-area. Local LFOs occur typically in the 1-2 Hz frequency range and appear only on small parts of the power system. Inter-area LFOs occur around the 0,1-1 Hz frequency range and exist in interconnected power systems through long tie lines [47]. The predictions were produced after 10^{-4} seconds.

Table 5: The table summarizes the methods used and the performances of the ML models associated with LFO problems.

Article	Year	Algorithm	Fault applied	Method	Task	Performance
[64]	2021	FFNN	Two scenarios: three-phase fault in weak tie and removal of load	Not mentioned	Predict power oscillation parameters	MSE = 0,0003195-0,0675
[65]	2018	FFNN	Single line outages and load variations	Static sliding window. Size = 0,067 s	Predict LFOs	MSE = 0,0052
					Predict localness of LFO: local or inter-area	All correctly classified
[66]	2017	Two RBFNN's	Single line outages and load variations	Not mentioned	Predict LFOs	MSE = 0,0172
					Classify critical generator	All correctly classified

8.1.4 Discussion about LFO models

The models in both articles produced great test times of around 10^{-4} seconds, which makes them suitable for real-time operation. In [65], the effect of having wind farms in a power system was examined. This could be of interest for Nordic TSOs and the NEWEPS project because the number of wind farms is going to increase soon, as discussed earlier in this thesis. No time window was mentioned in [64], while [65] operated with the first four cycles from the PMUs. Using a sampling rate of 60 Hz results in a 0,067 second time window. The authors in [65] used PCA to reduce the number of features in the dataset, going from 160 to 16. In larger systems, containing more PMUs, the number of features will increase. Utilizing PCA is therefore something that should be considered when building a predictive ML model. In [66], aside from predicting LFOs, the authors also determined the critical generator of the system. Critical generator meaning the generator responsible for creating the oscillations.

The authors argue that by quickly notifying the grid operator about where the critical generator is, corrective control actions can be performed, suppressing the oscillations in real-time. Including this in a decision support system gives the grid operators a more holistic view of the power system.

All the articles used an adequate number of samples in the datasets (>1200 samples). None of the articles presented the number of stable versus unstable samples in the dataset. Considering that the performance metric MSE was used, this does not pose a challenge, as long as the data samples did not contain too extreme outliers.

All the models produced great results. The test systems used were, however, rather small. This could explain the superior score from [64] as the author of this article used an 11-bus test system, while the authors of the other two articles both used a 39-bus test system. Each model's data was also artificially generated through simulation tools. Some properties from each model could, nonetheless, be beneficial in a future decision support system. Combining these properties in a hybrid model could produce a great real-time tool for assessing LFOs for grid operators.

8.2 Pre-fault models

8.2.1 Fault prediction

8.2.1.1 "PMU Analytics for Power Fault Awareness and Prediction"

In [67], the authors tested several ML algorithms to assess different tasks in the power system. The first task was to locate and classify single-line to ground faults and three-phase to ground faults, applied in the middle of the transmission lines. The faults lasted for 140 milliseconds. Multiple algorithms were tested, but the DT performed best. The second task was to forecast the same type of faults. The following ML algorithms were tested for this task: LR, SVR, MLP, and Gaussian Process Regression. LR and SVR performed best. For assessing the performance of the ML models, a ten-bus test system was used.

The inputs for the classification and forecasting models were voltage and current magnitudes and angles for all three phases from four PMUs. The measurement data is captured every 100 milliseconds from the data streams produced by the phasor measurement devices, resulting in a dataset containing 40 804 samples. This dataset was used to train and test classification models locating where the fault in the power system happened and classifying what kind of fault it was, either being a single line to ground faults or a three-phase to ground faults.

The forecasting models were able to predict the occurrence of electrical faults twelve timesteps (or 1200 milliseconds, one timestep = 100 milliseconds) ahead of the fault. The authors also found that by increasing the timesteps, the performance of the model decreased.

8.2.1.2 "Neural Network Based Early Warning System for an Emerging Blackout in Smart Grid Power Networks"

In [68], the authors proposed a method to predict whether or not a given power system is moving towards cascading failure. The method was divided into two phases. In the first phase, data from PMU was converted to probabilistic data by using probability distribution curves. Under normal power flow conditions, when all transmission lines are within their rated capacity, the curve has a symmetrical Gaussian distribution about its mean. As the system approaches cascading failures, the probability distribution curves become more and more non-Gaussian. This was used to label data points as being in normal condition or as tending towards a blackout, thus creating a historical dataset. The IEEE 30-bus test system was used to create the dataset, by tripping transmission lines and observing how the probability distribution changed. In the second phase, this dataset was used to train an FFNN. The four inputs for the FFNN were mean, variance, skewness, and kurtosis of the distribution curves. Once trained, the model could, according to the authors, be used in real-time operation to predict the critical transmission lines causing cascading failures.

The dataset consisted of 35 samples of normal and cascading situations. This was used for training and testing the FFNN.

Table 6: The table summarizes the methods used and the performances of the ML models associated with fault prediction.

Article	Year	Algorithm	Fault applied	Method	Task	Performance
[67]	2019	DT	Single and three-phase line to ground faults	Static sliding window: Size = 0,1 s	Detect fault	F1-score = 99,4 - 99,8%
					Classify fault	F1-score = 99,5%
		SVR			Predict faults 12-timesteps ahead	NRMSE = 9,26%
[68]	2015	FFNN	Tripping transmission lines randomly	Probability distribution curves	Predict cascading failures	MSE = $3{,}314*10^{-3}$
[69]	2018	FFNN	Tripping transmission lines randomly	Probability distribution curves	Predict cascading failures and provide counteractive measures	No performance given
[70]	2014	SVM with RBF kernel	Tripping highly loaded transmission lines	Probability distribution curves	Predict cascading failures	Acc = 100%

8.2.2 Discussion about fault prediction models

All the models, from both articles, [67] and [68], displayed great performances. The authors of [68] used a quite novel method to predict cascading failures in the power grid. Instead of using PMU data directly for training the model, the PMU data was used to create probability distribution curves, which in turn was used for training the FFNN model. The model was then able to give probabilities on how likely it is that one transmission line fault will produce a cascading failure.

In [69], the same authors built on this method and model and proposed a mitigation plan that made use of the early warning signals from the FFNN. The mitigation plan used forced line outage as a preventive measure against cascading failures. The line, which was being selected for the outage, was based on the maximum phase angle difference and grid topology. The reason for using phase angle differences as a measure was because it was observed that, right before the Northeast blackout of August 2003 in North America, the differences in phase angles showed a major increase. There is barely any literature about predictive ML systems using data from PMU with automatic counteractive measures built in. This contribution is therefore quite novel. This probabilistic approach was first presented in

[70] where an SVM was being used. The novelty of this method could be interesting to explore further.

The number of samples used for both training and testing in these articles was quite low (35 samples). Additionally, the authors did not use k-fold validation, which could have been used as a remedy for such a small dataset. As the models were based on artificially simulated data, a small dataset may produce a model that is not robust and does not generalize well when subjected to real-time data.

From Table 6 it can be seen that the F1-scores for the models in [67] were very high for detection and classification of faults. On the other hand, the model that tried to predict faults, scored a bit lower indicating that fault prediction is a more difficult task than classifying. This is in accordance with the assumptions made earlier in the thesis. The SVR model predicted 1200 milliseconds ahead of the fault which is not a lot of time for the grid operator to act on. But the model is still of high interest, as it assesses a challenge that is quite novel in this field, namely usage of PMU based ML models for fault prediction. Although not clearly stated in the article, it seemed like a prediction was made every 100 milliseconds. An interesting point discussed in the article was that the performance of the predictive model decreased as the timesteps increased. This problem must be addressed for the model to be more valuable for TSOs.

It should be noted that the test systems used in all the articles were rather small. The performance scores of the models might therefore decline when subjected to larger test systems.

8.2.3 Fault prediction models using non-PMU data

8.2.3.1 "Machine Learning to Predict Fault Events in Power Distribution Systems"

In [71], the author tested four different algorithms to predict faults in the grid based on expected weather data. The following four algorithms were tested: FFNN, SVM, DT, and Naïve Bayes. The trained models predicted if a fault was going to happen and where it was going to occur based on weather data. The research was divided into two parts. In the first part, the historical dataset was constructed for training the four models. F1-score was used to establish that the FFNN performed best. In the second part, the best-trained model was supposed to be utilized for real-time operation. This was, however, not done in the article.

The dataset, containing 3471 samples, was constructed by comparing historical fault events with the weather forecasts for the same period and the areas where the fault happened. The input for the models was weather data such as wind speed, humidity, and lightning strikes data.

8.2.3.2 *"Data-Based Line Trip Fault Prediction in Power Systems Using LSTM Networks and SVM"*

The authors in [72] proposed an LSTM and SVM hybrid model to predict transmission line trip faults in the power grid. Temporal information in the measurements was firstly caught by the three separate LSTM networks, one for each measurement. The measurements used for this model were current, voltage, and active power from every line. Then the features, captured by the LSTM networks, were merged and used as input for one SVM to predict if there was a fault or not. The most common reasons for line trip faults are, according to the authors, among others, aging, weather changes, and bad insulation. The authors argue that, before the line trip faults occur, there is a gradual indication in the line resistance. During this process, it is possible to detect a change in the electrical measurements. These changes are what the authors wanted to detect, thus being able to predict upcoming faults.

Real-world data was acquired from the China Southern Power Grid in the 2011-2014 period. The balanced dataset consisted of 5120 samples, where each sample contained 500 timesteps of current, voltage, and active power, 15 minutes before the event happened. The event was either a normal state or a fault. Data from 2011 to 2013 were used for training the model. The trained model was then used to predict faults for the first half of the year 2014. The training and testing times were around one minute.

Table 7: The table summarizes the methods used and the performances of the ML models associated with fault prediction using non-PMU data.

Article	Year	Algorithm	Fault applied	Method	Task	Performance
[71]	2021	FFNN	No faults applied, based on historical data	Not time-dependent data, no sliding window	Predict faults on components based on weather conditions	Avg. F1-score = 0,75
[72]	2018	LSTM and SVM hybrid	No faults applied, based on historical data	Predictions are given based on 15-minute real-time data	Predict faults in lines based on electrical measurements	Acc. = 95%
[73]	2017	FFNN	No faults applied, based on historical data	Prediction every 15 min, based on hourly new real-time data	Predicts line overloading based on multiple inputs	RMSE = 19,57-25,01
[74]	2017	LR	No faults applied, based on historical data	Not time-dependent data, no sliding window	Predict faults on components based on hurricane data	F1-score = 0,9027
[75]	2018	SVM	No faults applied, based on historical data	Not time-dependent data, no sliding window	Predict faults on components based on hurricane data	Avg. F1-score = 0,858

8.2.4 Discussion about fault prediction models using non-PMU data

The models in the articles performed to varying degrees. Articles [71], [74], and [75] were all mainly based on weather data and the location of the power system components. [71] used several weather conditions as input for the algorithms, providing a more holistic interpretation of the power system situation compared to the other two articles, which only used wind speeds as input. Real-world, historical data was being used for both training and testing the algorithms. This could explain the relatively low performance of these weather-based predictive models.

In [72], electrical measurements from transmission lines were used to predict faults in the grid. Although real-world, historical data was being used for training and testing, the trained model was able to predict with reasonably high accuracy. The hybrid model used by the authors should therefore be considered as a highly relevant model. Developing a system that uses several algorithms to solve different parts of the process, might be the way to go to achieve a well-performing system. The type of sensors that were being used to capture this data is not explained but is reasonable to think that a PMU could provide the data necessary. The test time was, however, not given but it is assumed that predictions are produced between 0 and 15 minutes before the fault happens, as the sampling period is 15 minutes. Considering that the authors made use of real-world data from a relatively large power system, this model and method are quite novel and should be further researched.

The authors in [73] used both weather data and grid data to predict ampacity levels in the three phases of transmission lines. The article used a quite novel method, considering the trained FFNN was being retrained in real-time, based on hourly new data. This has not been shown in any other article assessed in this thesis. Rather than just training the model once and let it predict on time-series data, the model is dynamic and adapts to changes in the grid. The model was tested using a simulator, producing a stream of time-dependent data. Predictions on the ampacity levels were produced every 15 minutes. The performance was poor, but the novelty of the method used should be remarked.

Chapter 9: Case study

This case study is included to illustrate how predictive ML models can learn from patterns in data from power systems. Just as the literature review, this case study is divided into pre- and post-fault.

9.1 Post-fault

9.1.1 Case description

For simulating a post-fault scenario, the test system created by Siemens called "savnw" was used. This is a 21-bus system with six generators. A one-line diagram of the system is represented in Figure 27. The generators of interest in this thesis are named G1 and G2 in the figure. As the goal of this case study is to assess how the generator angles respond in the different fault scenarios, a reference generator has to be chosen. Because generator G3 has the largest inertia in this test system, it is chosen as a reference. This is because the rotor angle of the generator with the largest inertia tends to move the least. To display how the system reacts to faults, the dynamic simulation function of PSS/E was used.

Three cases with two scenarios in each case were simulated. One where the system is able to restore stability and one where it becomes unstable and loses synchronism. To create these scenarios, a bus fault was applied one second into the simulation. This bus is marked with a black circle in the middle of Figure 27. Introducing a bus fault in a dynamic PSS/E simulation reduces the voltage at the faulted bus to almost zero, simulating a three-phase short-circuit fault in the same way as many of the articles presented above have done. The bus fault was resolved after 0,12 seconds (Case 1), 0,16 seconds (Case 2), and 0,08 seconds (Case 3). In a 50 Hz system, clearing a fault after 0,12 seconds results in a fault clearing time of six cycles. Graphs for Case 1 are presented in Figure 28 and Figure 29. Results of the three cases are given in Table 8. This table was produced through the Spyder integrated development environment using the programming language called Python. The code is given in the Appendix.

To simulate how such a fault could be resolved, power lines going to the faulted bus are tripped. In Figure 27, the blue cross indicates the line tripped in scenario 1 and the red crosses indicate the lines tripped in scenario 2. The total simulation run is 30 seconds. For each simulation the system is reset, assuring that the same start conditions apply to both scenarios. This was done for all three cases. More information about the test system is given in the Appendix.

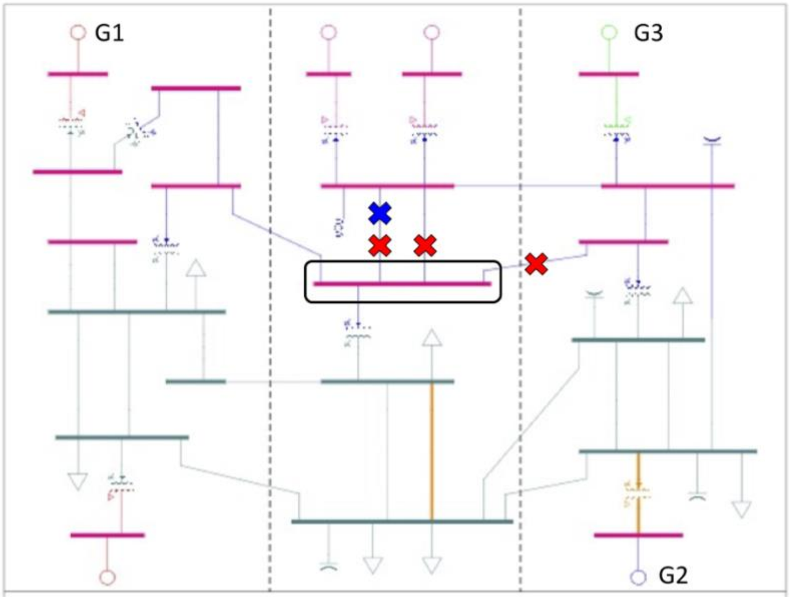
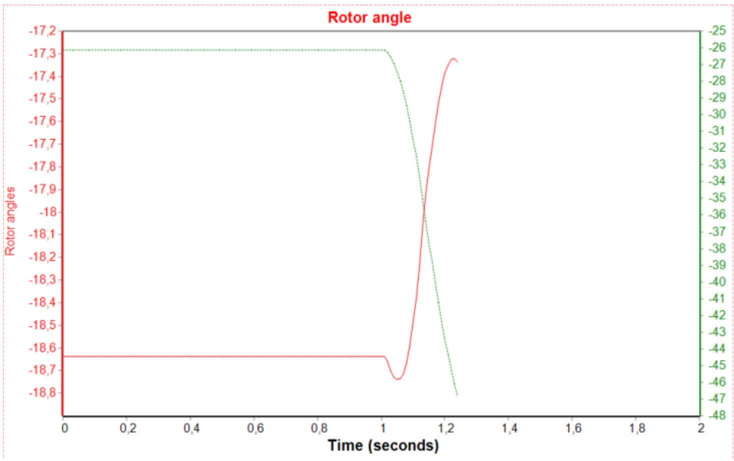


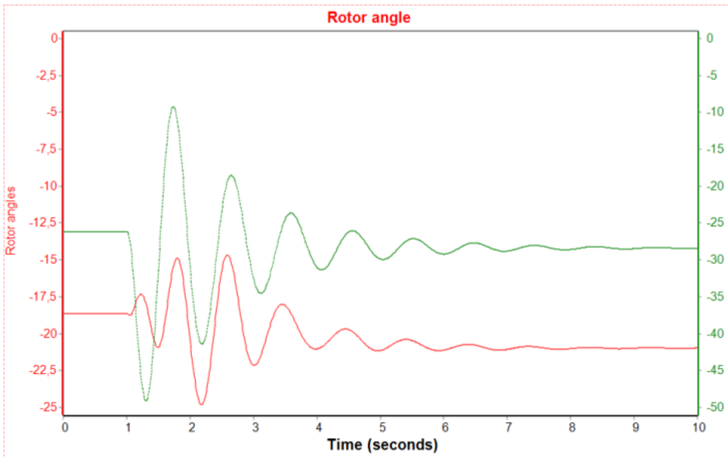
Figure 27: The example test system from Siemens called "savnw". The pink and green vertical lines are buses, the circles are generators, the two jagged lines are two winding transformers, the straight line parallel to a bent line are compensators, and the triangles are loads. In this thesis, G1 and G2 are analyzed, and G3 is used as the reference. The faulted bus is marked with a black square. The red and blue crosses indicate which lines are being tripped to resolve the fault.

9.1.2 Case results

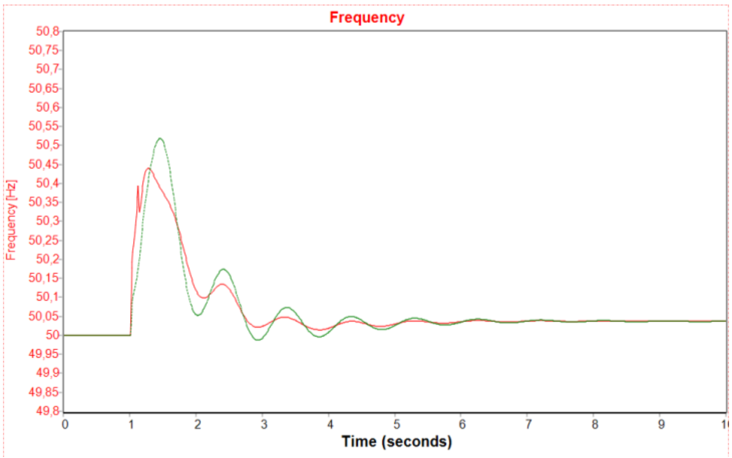
The results of all the simulations are presented below. Figure 28 and Figure 29 display the results from the stable and unstable situations for Case 1, respectively. The green line is G1, and the red line is G2, both with reference to G3. In both figures, **a)** displays the rotor angles 0,12 seconds after fault clearance (in Case 1 the fault was cleared 1,12 seconds into the simulation run), **b)** displays the rotor angles after 10 seconds of simulation, **c)** displays the frequency after 10 seconds of simulation and **d)** displays the rotor angles after 30 seconds of simulation.



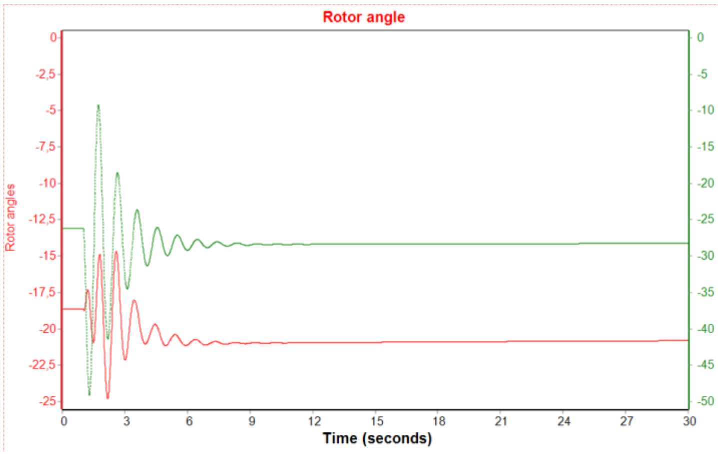
a)



b)



c)



d)

Figure 28: The figures display a stable situation for Case 1. The green line is G1, and the red line is G2. The rotor angles are measured in degrees and the frequency is measured in hertz. **a)** displays the rotor angles of the generators 0,12 seconds after fault clearance, relative to G3. **b)** displays the rotor angles of the generators ten seconds into the simulation, relative to G3. **c)** displays the frequency of the generators ten seconds into the simulation. **d)** displays the rotor angles of the generators 30 seconds into the simulation, relative to G3.

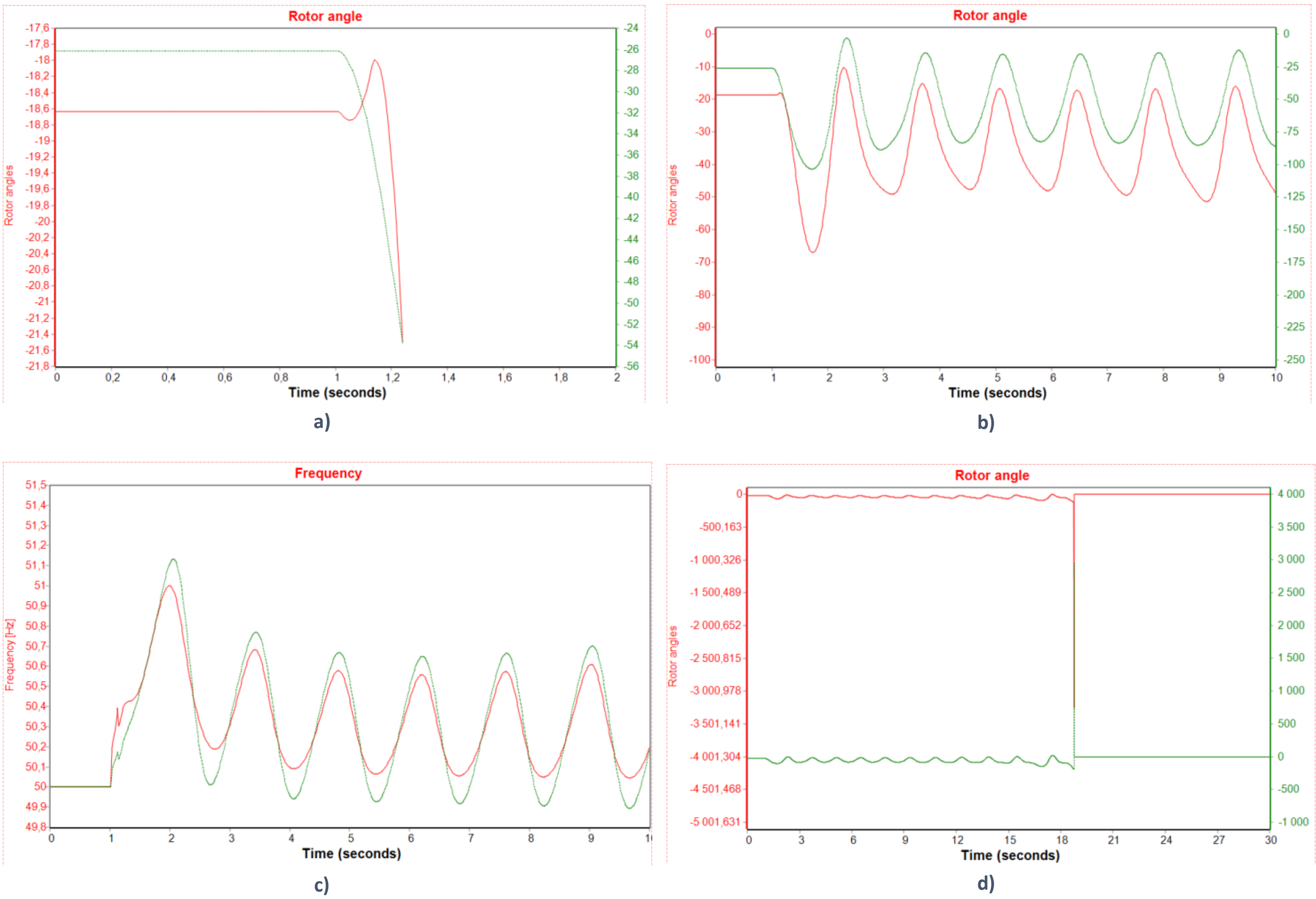


Figure 29: The figures display an unstable situation for Case 1. The green line is G1, and the red line is G2. The rotor angles are measured in degrees and the frequency is measured in hertz. **a)** displays the rotor angles of the generators 0,12 seconds after fault clearance, relative to G3. **b)** displays the rotor angles of the generators ten seconds into the simulation, relative to G3. **c)** displays the frequency of the generators ten seconds into the simulation. **d)** displays the rotor angles of the generators 30 seconds into the simulation, relative to G3.

Table 8 displays the results from the three cases and is an example of how a labeled dataset for training and testing of predictive ML algorithms could look like. The rightmost column contains the class label for each sample. A zero indicates a stable situation and a one indicates an unstable situation. The other columns contain timesteps after the instance of the fault being cleared. In this dataset, six cycles of post-fault rotor angle data are used as input for the ML algorithm. Each column represents one timestep, where each timestep is 0,02 seconds. The amount of post-fault data used in this dataset is inspired by multiple articles from the literature review.

Table 8: An example of a labeled dataset used for training and testing an ML algorithm. The table contains the post-fault rotor angles for the generators G1 and G2 relative to G3. Each case is either labeled as being stable or unstable. The columns represent time steps after the fault is cleared. This dataset contains six cycles of post-fault rotor angle data.

	0	0,02	0,04	0,06	0,08	0,10	Stable?
Case 1: G1 stable	-18,24	-17,93	-17,70	-17,52	-17,39	-17,33	0
Case 1: G2 stable	-33,70	-36,33	-38,84	-41,23	-43,40	-45,28	0
Case 2: G1 stable	-17,45	-17,01	-16,69	-16,47	-16,37	-16,40	0
Case 2: G2 stable	-39,55	-42,96	-46,12	-49,03	-51,58	-53,69	0
Case 3: G1 stable	-18,66	-18,59	-18,52	-18,46	-18,43	-18,41	0
Case 3: G2 stable	-29,47	-30,37	-32,20	-34,90	-36,59	-38,13	0
Case 1: G1 unstable	-18,24	-17,99	-18,11	-18,51	-19,21	-20,21	1
Case 1: G2 unstable	-33,70	-36,42	-39,49	-42,82	-46,34	-50,01	1
Case 2: G1 unstable	-17,45	-17,07	-17,11	-17,49	-18,23	-19,34	1
Case 2: G2 unstable	-39,55	-43,06	-46,83	-50,77	-54,81	-58,88	1
Case 3: G1 unstable	-18,66	-18,59	-18,73	-19,44	-20,21	-21,23	1
Case 3: G2 unstable	-29,47	-31,37	-33,70	-37,94	-39,55	-42,96	1

9.1.3 Case discussion

The figures in Figure 28 show that the power system quickly dampens the oscillations after only one second and is fully damped ten seconds into the simulation. The figures in Figure 29, on the other hand, show that the oscillations are not damped and continue to oscillate the whole simulation until around 20 seconds. Here the generators lose synchronism and spin out of control. The reason that the rotor angles return to zero shortly after the generators lose synchronism is because PSS/E trips the generators when the rotor angles exceed 180 degrees.

Figure 28 **a)** and Figure 29 **a)** reveal the generator angles 0,12 seconds after the fault is cleared. This is equivalent to six cycles of post-disturbance data, which was used by multiple authors from the post-fault category. In Figure 28 **a)** it is possible to see that the curve of G1, already at that early stage, resembles a damped oscillation. In Figure 29 **a)**, however, the peak of G1's curve happens earlier before it drops abruptly downwards. These patterns are what an ML model will learn when training on such data, gaining important information from data during a very short timeframe. This can be seen clearly in Table 8, where the differences in rotor angles between stable and unstable situations are displayed. A trained ML model, receiving this type of data, can compute predictions on stability after only 10^{-4} seconds as seen from the literature review.

To create an ML model that generalizes well, the dataset used for training and testing must contain a lot of different cases, with different scenarios and conditions. The dataset in Table 8 represents a small version of such a dataset. Creating a sufficient dataset could be done through power system software tools such as PSS/E.

9.2 Pre-fault

9.2.1 Case description and results

To display how ML models could potentially predict upcoming faults based on incipient faults and faults signatures, a case from Texas A&M will be used as an example. In this case, wildlife caused a fault on a transmission line on May 21. This is displayed in Figure 30. The fault was cleared by a fuse and the fuse was shortly thereafter replaced by service. Three days later, low-levels anomalies started to occur on the same phase of the same line. Figure 31 displays examples of these on May 24. The waveform seemed quite normal but on the third and fourth cycles, there were small distortions near the peaks. These subtle anomalies continued the next six days until May 30, when a customer reported lights-out. *"The responding crew found that the clamp connecting the primary phase conductor to the customer's service transformer had burned open"* [35]. The power outage on May 30 was a result of the wildlife fault nine days earlier which stressed the transmission line [35].

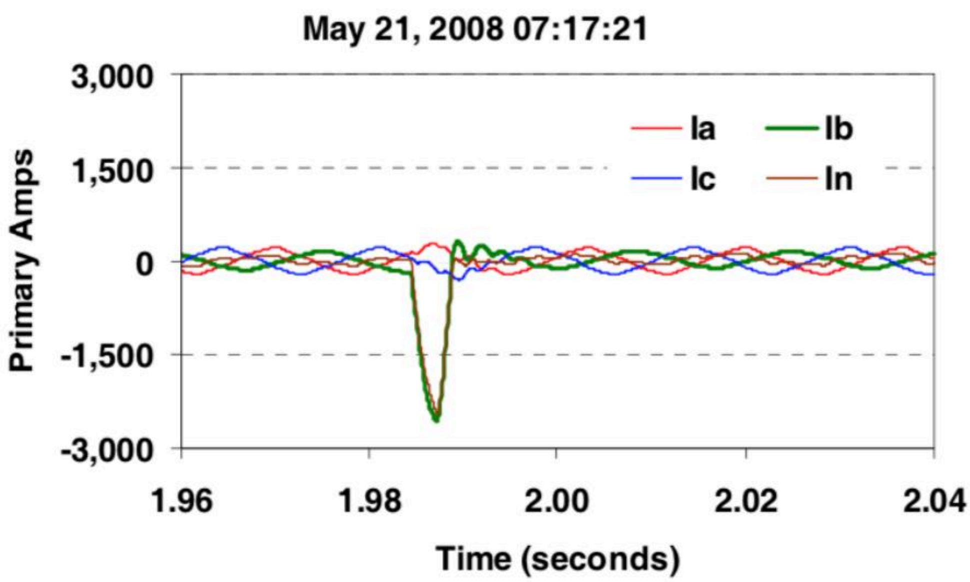


Figure 30: The figure displays a snapshot of the current signals right when the wildlife fault happened. From [35] [©2019 IEEE].

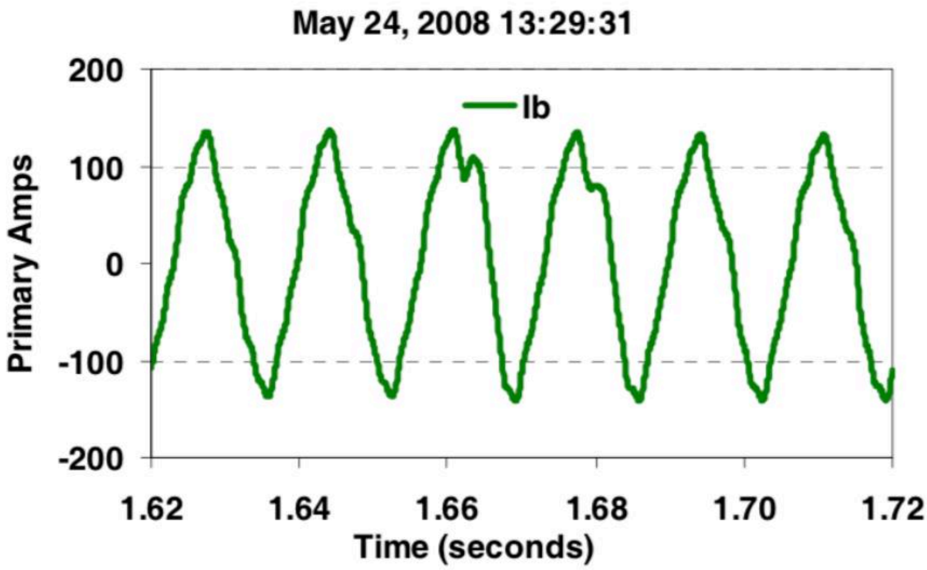


Figure 31: The figure displays a snapshot of the current signals three days later. It shows only phase b of the current signal. From [35] [©2019 IEEE].

9.2.2 Case discussion

This case displays that fault signatures can be very subtle and difficult for a human grid operator to register. Registering these signatures requires high-fidelity measuring devices. PMUs, measuring up to 60 samples per second, have a sampling rate capable of capturing such anomalies. As these devices produce large amounts of data every second, a data analysis method such as ML could be used. A trained ML model, such as an LR-based model from [67] or an RNN with LSTM-based model from [72], could be used to give early warnings to grid operators. In this case, an early warning system could have notified the grid operators about a recurrent anomaly and given a warning. This could in turn have prevented the final outage on May 30.

Chapter 10: Literature review discussion

10.1 Current usage of ML techniques in the Nordic synchronous area

Statnett's use of ML techniques for real-time fault and power system stability assessment is very limited. Through the conversation with an experienced grid operator at Statnett, it was found that there is an artificial intelligence-based program that aims at predicting faults in the grid. This is, however, not used by the operators because currently the conventional methods, which do not make use of predictive ML models, are faster and more reliable [59]. As mentioned earlier, SINTEF's EarlyWarn project has been researching ML models for fault prediction. Through the discussion with the representative, it was, however, determined that the research was in its early stages. Additionally, the models in this project mainly use data from PQAs as input. Research in this specific area has also been limited among the other Nordic TSOs. Through the Smart Transmission Grid Operation and Control (STRONGRID) project, the use of PMU data has been explored. ML techniques were, however, not tested.

The limited research and implementation of predictive ML models for fault and power system stability assessment done by the Nordic TSOs indicate a low TRL. Some research has been done, which results in a TRL of around 1 for both pre- and post-fault. The goal of the NEWEPS project is to create a prototype system. With such a system in place, the TRL will be at around 6. Although it is not given that the prototype system will be based on ML techniques, the TRL gives an indication on where the research and competence on predictive ML models at Statnett might be at the end of the project.

10.2 Post-fault

In this category, there has been done a substantial amount of research, especially with models predicting transient stability. For this specific task, the literature search resulted in four more articles than presented in Table 4. These were not prioritized due to time limitations. This does, however, give an indication of how much research there has been done on this topic. Articles concerning LFOs were, on the other hand, more difficult to discover. With the limitations set, only the articles given in Table 5 were found through the literature research.

To what extend is it possible to use Machine Learning models to predict the stability of the grid immediately after a fault happens?

The models reviewed in the literature study performed very well, having accuracies and RMSE values of around 95%-100% and 0,06-0,0003 respectively. The models are also fast enough to be used in real-time operation. None of the models are, however, tested on real-world data or in real-time operation. This will most probably decrease the performance of each model as mentioned earlier in this thesis. The articles reviewed are, nonetheless, great examples of the possibilities of predicting post-fault grid stability.

How relevant are the predictive Machine Learning models for Statnett and the NEWEPS project?

For the Nordic TSOs, the RNN using LSTM and the FFNN are two networks that perform very well and should be considered when starting research on ML models for post-fault stability assessment using PMU data. As gathering enough real-world data for training and testing models could be difficult, the hybrid simulator from [61] could be used in order to create realistic and non-misleading PMU data. To improve the robustness of the model, k-fold cross-validation could be implemented during training, especially when the dataset is small. A time-adaptive sliding window seems to produce slightly better scores and could also decrease the amount of post-fault data needed. It may, however, as mentioned, create a more complex model that has longer test times compared to a model using a static sliding window.

To decrease the training time of the models, PCA could be used to reduce the number of features. Besides predicting transient stability and LFOs, techniques for forecasting generator coherent groups are investigated which could help when performing controlled islanding of the grid. Models for both detecting critical generators causing the LFOs and determining the localness of LFO modes have also been presented. These features could be incorporated in a hybrid early warning system, such as NEWEPS.

Based on the literature review done in this thesis, this category of ML models is evaluated to currently be between TRL 3 and 4. The articles in this category display simulations handling realistic problems using representative datasets. As there has been done more research on transient stability compared to LFOs, ML models concerned with transient stability are evaluated to be at a TRL 4, while LFOs are evaluated to be at a TRL 3. As this is a substantial number of levels ahead of where the Nordic TSOs currently are, finding resources to improve ML models for stability prediction in post-fault situations should not be difficult.

10.3 Pre-fault

In this category, there has been a lot less research compared to the post-fault category, which resulted in dividing the research in this category into PMU models and non-PMU models. Every article found for ML models using data from PMUs, meeting the set limitations, is presented in Table 6. For non-PMU models, in Table 7, there were two more articles found which met the limitations, but these were not prioritized. It should be noted that three of the articles concerning PMU based models discuss the same method and model, decreasing the amount of relevant literature even further.

To what extend is it possible to use Machine Learning models to predict upcoming faults in the grid before the fault happens?

The overall performance of the ML models in this category is slightly worse than for post-fault models, as displayed in the tables presented in Chapter 8. This indicates a difficulty in predicting upcoming faults which could be a reason for the limited literature on this topic. In all the articles using data from PMUs, the datasets were simulated. A reason for this could be the scarcity of relevant real-world PMU data and the convenience of using simulation tools. When subjected to real-world data, ML models trained solely on artificially simulated data could experience a decrease in performance. This can especially happen when the models are tested on small test systems or trained on small datasets. In the articles concerning ML models employing non-PMU data, real-world historical data was, however, adopted. This increases the relevancy and the generalization ability of these models. The articles reviewed in this category display that using ML models to predict upcoming faults could be possible in the future. The research done on these ML models using data from PMUs is, however, in its beginning stages.

How relevant are the predictive Machine Learning models for Statnett and the NEWEPS project?

For the NEWEPS project, the results from this category could be of high interest as this is a topic where there has been limited research and testing. Especially the hybrid model (RNN using an LSTM with SVM classifier) from [72] is very relevant as it is used on real-world data and performs very well. Using one LSTM network for each measurement could be an intelligent way to incorporate more information into the predictive model. Although the model does not use PMU data, adjusting the method used in [72] will most probably make it capable of using PMU data.

The model in [67] uses PMU data as input but has a slightly worse score than [72] despite the authors using a smaller test system in [67]. The article is, however, an important contribution as it this is the only model that uses PMU data directly in this category. A dynamic model that adapts to changes in the grid as shown in [73], should also be evaluated and tested as this will result in a more flexible predictive system. Incorporating weather data in a predictive early warning system could be beneficial and give a more holistic and comprehensive view of the power grid. As weather conditions have been shown to have a large impact and correlation on the number of faults and disturbances happening in the grid, these data should be included in such a system.

This category of ML models is evaluated to currently be between a TRL 3 and 4. Articles about predictive ML models using data from PMUs are quite limited indicating a low TRL. Some research in this specific area has, however, been done resulting in a TRL 3. By including ML models using non-PMU data the amount of research done increases. [72] displays experiments on real-world power systems indicating a higher TRL. This specific category is therefore evaluated to be at TRL 4. Through the NEWEPS project, the Nordic TSOs are in a great position to be part of this important research of using ML to predict faults and disturbances in the grid.

10.4 Benefits of predictive ML models

Creating and implementing an early warning system would yield benefits viewing it from both an economic as well as an operational standpoint. Operating the grid in a more probabilistic way, and thus moving further away from the deterministic N-1 concept, which was presented earlier in the thesis, could reduce costs for the Nordic TSOs. An example of this is reduced material costs when building new power lines, as fewer lines might be needed. Such a system could potentially also reduce the costs associated with UE, as more of the faults and disturbances happening in the grid could be anticipated.

A reliable early warning system could help grid operators in keeping an increasingly more dynamic and complex power system stable and preventing outages. More extreme weather conditions will most likely make this task even more difficult. Fault prediction will therefore be an important part of such an early warning system.

Utilizing predictive ML models as presented in this thesis could be an important step in creating such an early warning system and realizing the benefits mentioned above. This system could be divided into two separate subsystems, one focusing on analyzing fault signatures and predicting faults and disturbances, in the pre-fault timeframe and the other one focusing on predicting system stability, in the post-fault timeframe.

Chapter 11: Conclusion

This master's thesis has given an overview of the research and the current status of ML models using PMU data for predicting power system stability and faults. Through a literature study, it was decided that the state-of-the-art algorithms should be divided into either concerning pre- or post-fault problems. Such a categorization has not been observed in the literature and is therefore unique for this master's thesis. The thesis highlights the important distinction between these two timeframes. By reviewing the relevant literature, meeting the set limitations in this thesis, it has been found that the current usage of such models is quite limited and is still in its early stages. The model testing performed in these articles does, however, display promising results, indicating that predictive ML prototypes using PMU data may not be far from being developed.

This thesis has found that there is more literature and more research done on post-fault models using data from PMUs than on pre-fault. Including only ML models using data from PMUs as input, the models belonging to the post-fault category are evaluated to be between a TRL 3 and 4, while pre-fault models are evaluated to have a TRL of 3. By including non-PMU models, the pre-fault category is evaluated to have a TRL of 4. A valuable overview of the most common ML models researched and the expected performance, displayed as tables, has in this thesis been produced. Different methods for using the data from PMUs have been explained, presented, and evaluated. By comparing the speed, performance, and relevancy of each model, recommendations have been made for the Nordic TSOs to further pursue. The information and overview presented in this thesis could be used as a starting point for further research.

In the case study, it was displayed through examples how ML models can predict stability and faults in the power grid. For the post-fault timeframe, examples were created using the dynamic simulation function of the power system software tool PSS/E. A three-phase short-circuit fault was applied to create two different scenarios in three different cases. The fault, the fault clearing time, and the amount of post-fault data used in this simulation were based on the results from the literature review. Through the three cases, it was found that there are differences in the rotor angles of the generators after only 0,08 to 0,16 seconds after the fault is cleared, comparing the stable and the unstable cases. This shows that there are clear patterns which an ML model can learn, and thus differentiate between a stable and an unstable case. For the pre-fault timeframe, a case from Texas A&M was used. It was found that there are subtle anomalies in the waveforms ahead of certain faults. These are so subtle that they are difficult for human grid operators to detect. Trained ML models, however, have the capability of detecting such small anomalies, and thus being able to give early warnings about upcoming faults.

A TRL evaluation of the current state of predictive ML models using PMU data by Statnett and the other Nordic TSOs has also been carried out. This has been done through conversations with relevant experts in the fields of ML and power system operation. Both the pre- and post-fault categories have been evaluated to be at TRL of 1, seeing that there has almost not been done research on this specific topic. The results from this thesis could therefore prove to be a great entry for starting this specific research for the NEWEPS project, as the project's goal is to reach a TRL of 6.

The discussion, based on the results from the literature review, culminated in recommendations of the most promising algorithms and methods. More research on these algorithms and methods could prove to be beneficial. For stability assessment in post-fault situations, an RNN with LSTM or an FFNN seems to produce the best scores. Using a time-adaptive sliding window might help to decrease the test times but could also make the model more complex. For fault forecasting in pre-fault situations using data from PMUs, a SVR seems to be the most promising model. However, by including predictive ML models using non-PMU, a hybrid model using RNN with LSTM and an SVM classifier might be the better option as this has shown superior performance compared to the SVR.

Other predictive models using non-PMU data have also been assessed to compliment the lack of PMU based predictive models. The use of weather data as an additional input to an ML model has been discussed and it has been considered to be a great way of producing a model that gives a more holistic real-time view of the power grid. This would increase the number of features that the model would need to learn. PCA has, therefore, been suggested as a counteractive measure to decrease the number of features without losing valuable information.

The benefits, for Statnett and the other Nordic TSOs, of using a predictive ML model in an early warning system has also been discussed, seeing it both from an economical and an operational point of view. Implementing ML models could potentially decrease expenses associated with planning and constructing new power lines, and with expenses because of UE. Predictive ML models could also be used as a decision support tool, assisting grid operators with keeping the power grid stable.

11.1 Further work and recommendations

Through the research done in this master's thesis, it has been found that predictive ML models using PMU data are highly relevant for the NEWEPS project and should be further researched. The following algorithms are the most promising found in this thesis:

- For quick prediction of power grid stability in the post-fault timeframe: RNN with LSTM or FFNN
- For fault prediction in the pre-fault timeframe: RNN with LSTM and SVM hybrid

Further research on these algorithms should be done, increasing Statnett's TRLs in both categories. The models should be tested on relevant data and simulations of the Nordic power grid. This master's thesis could be a great starting point for getting an overview of the different simulation methods used. The case study could be used as inspiration for creating larger simulations. Incorporating real-world data would be beneficial to create a more robust model. Creating a prototype system with predictive ML models implemented is the next step after simulation testing. These prototypes should be tested in real-time and real-world operation, eventually becoming an essential part of normal grid operation for Statnett.

Bibliography

[1] Our World In Data, "Global direct primary energy consumption," *Our World In Data*. https://ourworldindata.org/grapher/global-primary-energy?country=~OWID_WRL (accessed Apr. 03, 2021).

[2] R. Benestad, J. Mamen, K. Harstveit, and J. S. Fuglestevdt, "klimaendringer i Store norske leksikon," *Store norske leksikon*, 2021. https://snl.no/klimaendringer (accessed Apr. 03, 2021).

[3] European Commission, "A European Green Deal," *European Commission*, 2019. https://ec.europa.eu/info/strategy/priorities-2019-2024/european-green-deal_en (accessed Feb. 02, 2021).

[4] CINELDI, "Annual Report 2019," 2019.

[5] K. Uhlen, P. Overholt, and O. Valentine, "Synchrophasor applications for wide area monitoring and control," *ISGAN Discuss. Pap.*, vol. Task 1, 2016.

[6] B. Yang, J. Yamazaki, N. Saito, Y. Kokai, and D. Xie, "Big data analytic empowered grid applications - Is PMU a big data issue?," IEEE Computer Society, Aug. 2015. doi: 10.1109/EEM.2015.7216718.

[7] S. Azad, F. Sabrina, and S. Wasimi, "Transformation of smart grid using machine learning," *29th Australas. Univ. Power Eng. Conf. AUPEC 2019*, 2019, doi: 10.1109/AUPEC48547.2019.211809.

[8] C. A. Andresen, B. N. Torsæter, H. Haugdal, and K. Uhlen, "Fault Detection and Prediction in Smart Grids," in *2018 IEEE 9th International Workshop on Applied Measurements for Power Systems (AMPS)*, Sep. 2018, pp. 1–6, doi: 10.1109/AMPS.2018.8494849.

[9] Statnett SF, "Nordic Early Warning Early Prevention System (NEWEPS)," 2020, no. March, pp. 1–13.

[10] Energifakta Norge, "Strømnettet," *Energifakta Norge*, 2019. https://energifaktanorge.no/norsk-energiforsyning/kraftnett/ (accessed Feb. 08, 2021).

[11] Energifakta Norge, "Forsyningssikkerhet," *Energifakta Norge*, 2019. https://energifaktanorge.no/norsk-energiforsyning/forsyningssikkerhet/ (accessed Feb. 08, 2021).

[12] ENTSO-E, "First milestone of Future Synchronous Connection of the Baltic Power System with Continental Europe," *ENTSO-E*, 2019. https://www.entsoe.eu/news/2019/05/29/first-milestone-of-future-synchronous-connection-of-the-baltic-power-system-with-continental-europe/ (accessed Feb. 08, 2021).

[13] S. Berlijn, "FYS377 lecture by Sonja Berlijn - Kraftsystemets rolle." 2020.

[14] L. Mantzos, T. Wiesenthal, F. Neuwahl, and M. Rózsai, "The POTEnCIA Central scenario: An EU energy outlook to 2050," 2019. doi: 10.2760/32835.

[15] ENTSOE-E, "Ten-Year Network Development Plan 2020 Highlights," 2020.

[16] WindEurope, "Breaking new ground - Wind Energy and the Electrification of Europe's Energy System," pp. 40–43, 2018.

[17] The Nordic TSOs, "Challenges and Opportunities for the Nordic Power System," 2016.

[18] The Nordic TSOs, "Nordic Grid Development Plan 2019," no. June, 2019.

[19] Norsk Elbilforening, "Elbilsalg," *Norsk Elbilforening*, 2020. https://elbil.no/elbilstatistikk/elbilsalg/ (accessed Feb. 02, 2021).

[20] NVE, “Syv elektrifiseringstiltak kan kutte 2,3 millioner tonn CO2 i industrien,” *NVE*, Jun. 26, 2020. https://www.nve.no/nytt-fra-nve/nyheter-energi/syv-elektrifiseringstiltak-kan-kutte-2-3-millioner-tonn-co2-i-industrien/ (accessed Feb. 11, 2021).

[21] M. Panteli and P. Mancarella, “Influence of extreme weather and climate change on the resilience of power systems: Impacts and possible mitigation strategies,” *Electr. Power Syst. Res.*, vol. 127, p. 260, 2015, doi: 10.1016/j.epsr.2015.06.012.

[22] ENTSOE-E, “Research, Development & Innovation Roadmap 2020 – 2030,” 2020.

[23] J. D. Glover, T. J. Overbye, and M. S. Sarma, *Power System Analysis & Design*, 6. Boston: Cengage Learning, 2016.

[24] Electric Power Group and CERTS, “Phasor Technology and Real-Time Dynamics Monitoring System (RTDMS),” 2006.

[25] A. G. Phadke and J. S. Thorp, “Phasor Representation of Sinusoids,” in *Synchronized Phasor Measurements and Their Applications*, 2., J. H. Chow, A. M. Stankovic, and D. Hill, Eds. Springer International Publishing, 2008, pp. 5–6.

[26] R. F. Nuqui, “State Estimation and Voltage Security Monitoring Using Synchronized Phasor Measurements [Dissertation],” Virginia, 2001.

[27] H. Bentarzi, M. Tsebia, and A. Abdelmoumene, “PMU based SCADA enhancement in smart power grid,” *2018 IEEE 12th Int. Conf. Compat. Power Electron. Power Eng. (CPE-POWERENG 2018)*, pp. 1–6, 2018, doi: 10.1109/CPE.2018.8372580.

[28] U.S. Energy Information Administration (EIA), “New technology can improve electric power system efficiency and reliability - Today in Energy,” *U.S. Energy Information Administration (EIA)*, 2012. https://www.eia.gov/todayinenergy/detail.php?id=5630 (accessed Apr. 03, 2021).

[29] O. Finseth, “Personal Communication about PMUs [26.03.].” Unpublished, 2021.

[30] S. Raschka and V. Mirjalili, *Python Machine Learning*, 3rd ed., no. December 2019. Birmingham, 2019.

[31] J. Brownlee, *Deep Learning With Python: Develop Deep Learning Models On Theano And TensorFlow Using Keras*, 1.8. Melbourne: Machine Learning Mastery, 2019.

[32] K. Solberg, “Online Voltage Instability Detection and Prevention Using PMUs - a NEWEPS Perspective,” NMBU, 2021.

[33] K. W. Hoiem, V. Santi, B. N. Torsater, H. Langseth, C. A. Andresen, and G. H. Rosenlund, “Comparative Study of Event Prediction in Power Grids using Supervised Machine Learning Methods,” in *2020 International Conference on Smart Energy Systems and Technologies (SEST)*, Sep. 2020, pp. 1–6, doi: 10.1109/SEST48500.2020.9203025.

[34] Siemens Industry Inc., “Program Operation Manual PSS/E 34.6.1,” 2019.

[35] 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,” *2009 IEEE Power Energy Soc. Gen. Meet. PES ’09*, pp. 1–8, 2009, doi: 10.1109/PES.2009.5275415.

[36] J. Brownlee, “Difference Between Classification and Regression in Machine Learning,” *Machine Learning Mastery*, 2017. https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/ (accessed Feb. 25, 2021).

[37] M. Sunasra, “Performance Metrics for Classification problems in Machine Learning,” *Medium*, 2017. https://medium.com/@MohammedS/performance-metrics-for-

classification-problems-in-machine-learning-part-i-b085d432082b (accessed Mar. 05, 2021).
[38] A. Mishra, “Metrics to Evaluate your Machine Learning Algorithm,” *Towards Data Science*, 2018. https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234 (accessed Mar. 05, 2021).
[39] S. Sreenivasa, “Radial Basis Function (RBF) Kernel: The Go-To Kernel,” *Towards Data Science*, 2020. https://towardsdatascience.com/radial-basis-function-rbf-kernel-the-go-to-kernel-acf0d22c798a (accessed Mar. 23, 2021).
[40] P. Gupta, “Decision Trees in Machine Learning,” *Towards Data Science*, 2017. https://towardsdatascience.com/decision-trees-in-machine-learning-641b9c4e8052 (accessed Mar. 01, 2021).
[41] T. Ahadli, “Most Effective Way To Implement Radial Basis Function Neural Network for Classification Problem,” *Towards Data Science*, 2020. https://towardsdatascience.com/most-effective-way-to-implement-radial-basis-function-neural-network-for-classification-problem-33c467803319 (accessed Apr. 03, 2021).
[42] J. Brownlee, *Deep Learning for Time Series Forecasting*, 1.4. Melbourne: Machine Learning Mastery, 2018.
[43] J. L. Viegas, S. M. Vieira, R. Melicio, H. A. Matos, and J. M. C. Sousa, “Prediction of events in the smart grid: Interruptions in distribution transformers,” *Proc. - 2016 IEEE Int. Power Electron. Motion Control Conf. PEMC 2016*, pp. 436–441, 2016, doi: 10.1109/EPEPEMC.2016.7752037.
[44] R. Zhang, Y. Xu, Z. Y. Dong, and K. P. Wong, “Post-disturbance transient stability assessment of power systems by a self-adaptive intelligent system,” *IET Gener. Transm. Distrib.*, vol. 9, no. 3, pp. 296–305, 2015, doi: 10.1049/iet-gtd.2014.0264.
[45] W. Sattinger, H. Al-Zahrani, G. Giannuzzi, S. Hur, and H. Khazraj, “Power System Operation and Control: Wide area monitoring systems- Support for control room applications,” 2018.
[46] P. Kundur *et al.*, “Definition and Classification of Power System Stability,” *IEEE Trans. Power Syst.*, vol. 19, no. 3, pp. 1387–1401, 2004, doi: 10.1109/pes.2005.1489266.
[47] K. Prasertwong, N. Mithulananthan, and D. Thakur, “Understanding Low-Frequency Oscillation in Power Systems,” *Int. J. Electr. Eng. Educ.*, vol. 47, no. 3, pp. 248–262, 2010, doi: 10.7227/IJEEE.47.3.2.
[48] A. S. Masoum, N. Hashemnia, A. Abu-Siada, M. A. S. Masoum, and S. M. Islam, “Online Transformer Internal Fault Detection Based on Instantaneous Voltage and Current Measurements Considering Impact of Harmonics,” *IEEE Trans. Power Deliv.*, vol. 32, no. 2, pp. 587–598, 2017, doi: 10.1109/TPWRD.2014.2358072.
[49] Statnett SF: Avdeling Feilanalyse, “Årsstatistikk 2018 Driftsforstyrrelser og feil i 33-420 kV-nettet,” Oslo, 2019.
[50] G. Kjølle, “KILE - Kvalitetsjusterte inntektsrammer ved ikke levert energi,” *SINTEF*. https://www.sintef.no/ekspertise/sintef-energi/kile-kvalitetsjusterte-inntektsrammer-ved-ikke-lev/ (accessed Mar. 28, 2021).
[51] Elprocus, “Types of Faults in Electrical Power Systems and Their Effects,” *Elprocus*. https://www.elprocus.com/what-are-the-different-types-of-faults-in-electrical-power-systems/ (accessed Mar. 29, 2021).
[52] Circuit Globe, “What are the Different Types of Faults in Power System?,” *Circuit Globe*. https://circuitglobe.com/types-of-faults-in-power-system.html (accessed Mar.

29, 2021).
[53] T. S. Salem, K. Kathuria, H. Ramampiaro, and H. Langseth, "Forecasting intra-hour imbalances in electric power systems," *33rd AAAI Conf. Artif. Intell. AAAI 2019, 31st Innov. Appl. Artif. Intell. Conf. IAAI 2019 9th AAAI Symp. Educ. Adv. Artif. Intell. EAAI 2019*, pp. 9595–9600, 2019, doi: 10.1609/aaai.v33i01.33019595.
[54] Statnett SF, "IMPALA Management Summary," Oslo, 2015. doi: 10.1007/978-3-662-45832-7_1.
[55] E. Lindeberg, "Discussion about IMPALA [Personal communication, 26.03.]." Unpublished, 2021.
[56] Statnett SF, "Ny modell for balansering av det nordiske kraftsystemet med kvartersoppløsning," *Statnett SF*, 2018. https://www.statnett.no/for-aktorer-i-kraftbransjen/utvikling-av-kraftsystemet/prosjekter-og-tiltak/finere-tidsopplosning-i-det-nordiske-kraftmarkedet/ (accessed Mar. 29, 2021).
[57] SINTEF, "EarlyWarn," *SINTEF*, 2017. https://www.sintef.no/en/projects/2017/earlywarn/ (accessed Mar. 29, 2021).
[58] C. A. Andresen, "Discussion about EarlyWarn [Personal communication 28.01.]." Unpublished, 2021.
[59] G. Sætermoe, "Discussion about grid operation [Personal communication, 12.03.]." Unpublished, 2021.
[60] S. A. Siddiqui, K. Verma, K. R. Niazi, and M. Fozdar, "Real-Time Monitoring of Post-Fault Scenario for Determining Generator Coherency and Transient Stability Through ANN," *IEEE Trans. Ind. Appl.*, vol. 54, no. 1, pp. 685–692, Jan. 2018, doi: 10.1109/TIA.2017.2753176.
[61] T. Behdadnia, Y. Yaslan, and I. Genc, "A new method of decision tree based transient stability assessment using hybrid simulation for real-time PMU measurements," *IET Gener. Transm. Distrib.*, vol. 15, no. 4, pp. 678–693, 2021, doi: 10.1049/gtd2.12051.
[62] S. Jafarzadeh and V. M. I. Genc, "Real-time transient stability prediction of power systems based on the energy of signals obtained from PMUs," *Electr. Power Syst. Res.*, vol. 192, p. 107005, Mar. 2021, doi: 10.1016/j.epsr.2020.107005.
[63] J. J. Q. Yu, D. J. Hill, A. Y. S. Lam, J. Gu, and V. O. K. Li, "Intelligent Time-Adaptive Transient Stability Assessment System," *IEEE Trans. Power Syst.*, vol. 33, no. 1, pp. 1049–1058, Jan. 2018, doi: 10.1109/TPWRS.2017.2707501.
[64] N. Ca, R. Satheesh, and S. Rajan, "Prediction of Electromechanical Oscillatory Parameters in Power Systems Using ANN," 2021, doi: 10.1109/ICEPE50861.2021.9404526.
[65] A. K. Gupta, K. Verma, and K. R. Niazi, "Wide-area PMU-ANN based monitoring of low frequency oscillations in a wind integrated power system," in *2018 8th IEEE India International Conference on Power Electronics (IICPE)*, Dec. 2018, pp. 1–6, doi: 10.1109/IICPE.2018.8709466.
[66] A. K. Gupta, K. Verma, and K. R. Niazi, "Intelligent wide area monitoring of power system oscillatory dynamics in real time," in *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*, Jan. 2017, pp. 1–6, doi: 10.1109/ICACCS.2017.8014574.
[67] M. Z. Che Wanik, A. Sanfilippo, N. Singh, A. Jabbar, and Z. Cen, "PMU Analytics for Power Fault Awareness and Prediction," in *2019 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA)*, May 2019, pp. 1–8, doi: 10.1109/SGSMA.2019.8784461.

[68] S. Gupta, F. Kazi, S. Wagh, and R. Kambli, “Neural Network Based Early Warning System for an Emerging Blackout in Smart Grid Power Networks,” in *Intelligent Distributed Computing. Advances in Intelligent Systems and Computing, vol 321*, R. Buyya and S. Thampi, Eds. Springer, Cham, 2015, pp. 173–183.
[69] S. Wagh, F. Kazi, S. Gupta, and N. Singh, “Augmenting WAMPAC with machine learning tools for early warning and mitigation of blackout events,” *Int. J. Humanit. Technol.*, vol. 1, no. 1, p. 83, 2018, doi: 10.1504/IJHT.2018.10011336.
[70] S. Gupta, R. Kambli, S. Wagh, and F. Kazi, “Support-Vector-Machine-Based Proactive Cascade Prediction in Smart Grid Using Probabilistic Framework,” *IEEE Trans. Ind. Electron.*, vol. 62, no. 4, pp. 2478–2486, Apr. 2015, doi: 10.1109/TIE.2014.2361493.
[71] A. Dagnino, “Machine Learning to Predict Fault Events in Power Distribution Systems,” in *Data Analytics in the Era of the Industrial Internet of Things*, Springer, Cham, 2021, pp. 47–60.
[72] S. Zhang, Y. Wang, M. Liu, and Z. Bao, “Data-Based Line Trip Fault Prediction in Power Systems Using LSTM Networks and SVM,” *IEEE Access*, vol. 6, pp. 7675–7686, 2018, doi: 10.1109/ACCESS.2017.2785763.
[73] R. Fainti, M. Alamaniotis, and L. H. Tsoukalas, “Three-phase line overloading predictive monitoring utilizing artificial neural networks,” in *2017 19th International Conference on Intelligent System Application to Power Systems (ISAP)*, Sep. 2017, pp. 1–6, doi: 10.1109/ISAP.2017.8071397.
[74] R. Eskandarpour and A. Khodaei, “Machine Learning Based Power Grid Outage Prediction in Response to Extreme Events,” *IEEE Trans. Power Syst.*, vol. 32, no. 4, pp. 3315–3316, Jul. 2017, doi: 10.1109/TPWRS.2016.2631895.
[75] R. Eskandarpour, A. Khodaei, and A. Arab, “Improving power grid resilience through predictive outage estimation,” *2017 North Am. Power Symp.*, pp. 1–5, 2017, doi: 10.1109/NAPS.2017.8107262.

Appendix

A. TRL table

Table 9: The table displays the different TRLs and explains each level thoroughly. From Statnett.

TRL		Description software	Description methods/knowledge
1	**Basic principles are observed**	Literature studies are conducted to confirm basic principles for the concept. A possible idea is described which is based on the identified principles. This also includes mathematical formulations for possible algorithms.	Literature studies are conducted to confirm basic principles of the technology. A possible idea is described which is based on the identified principles.
2	**Concepts are formulated**	Practical application is formulated which includes functional requirements, system limits and whether the software is to be part of existing software or independent. Preparatory analytical studies with synthetic data support the concept and the basic system architecture with modules/functions is identified.	Practical application of the method or need for new knowledge is formulated. This also includes the delimitation of the problem area. Preparatory analytical studies support the concept and the concept for obtaining new knowledge/data is described.
3	**The critical functions or characteristics of the concept are proven**	Active work with the software has started. This includes coding of limited functions to validate critical properties and analytical assumptions. All testing takes place with non-integrated software components and partially representative data.	Active work on the concept has begun. This includes analytical studies to prove that the method can work or to be able to confirm that the chosen concept for obtaining new knowledge/data is suitable.
4	**Technological component or concept is developed**	Software components are integrated into a first trial version to determine their compatibility. Basic functionality is tested in simplified environments. Requirements for data quality and format are formulated.	Basic parts of the new method are developed and adapted to the need. The concept for obtaining new knowledge/data is developed in detail so that it is certain that it can be used.
5	**Technological component or concept is validated in relevant environment**	The software components are integrated so that the system configuration can be tested. Experiments with realistic problems, representative data sets and simulated interfaces to existing systems are carried out. Coding of functions/modules is completed and software testing is started.	All parts of the new method are integrated into a system to determine that they work together. The method is tested with realistic issues. The concept of obtaining new knowledge/data is tested in a very limited area/sample to be able to see if it works.
6	**Prototype or concept demonstrated in relevant environment**	The practical use and performance of the software solution is demonstrated with a complete prototype, ready-made user interface and realistic issues. The software has all the features but is likely to contain a number of known or unknown bugs. During the demonstration, the software is partially integrated with existing systems.	The practical feasibility of the new method is demonstrated with realistic issues. The concept for obtaining new knowledge/data is tested in a limited area/sample by going through the entire process which also includes analysis of data and the like.
7	**Full-scale prototype or concept demonstrated in operating environment**	The prototype of the software solution is implemented in an operational environment, where critical functionality is available for demonstration. The software version is ready for use if there are no significant errors. During the demonstration test, the software is fully integrated with operating systems.	The new method is demonstrated in an operational environment, where it is integrated with existing operational solutions and processes. New knowledge/data is obtained and analyzed so that it can be used in operational processes.
8	**Qualification in operational environment**	The software is fully integrated with operating systems and all functionality is tested in use. Documentation of the software is completed.	The method/knowledge is used and evaluated in real operational situations.
9	**Used in real, operative situation**	The software is used in its final form in operation over a longer period of time. All documentation is verified and long-term software support is in place.	The method/knowledge is used in its final form in operation over a longer period of time.

B. More information about PSS/E

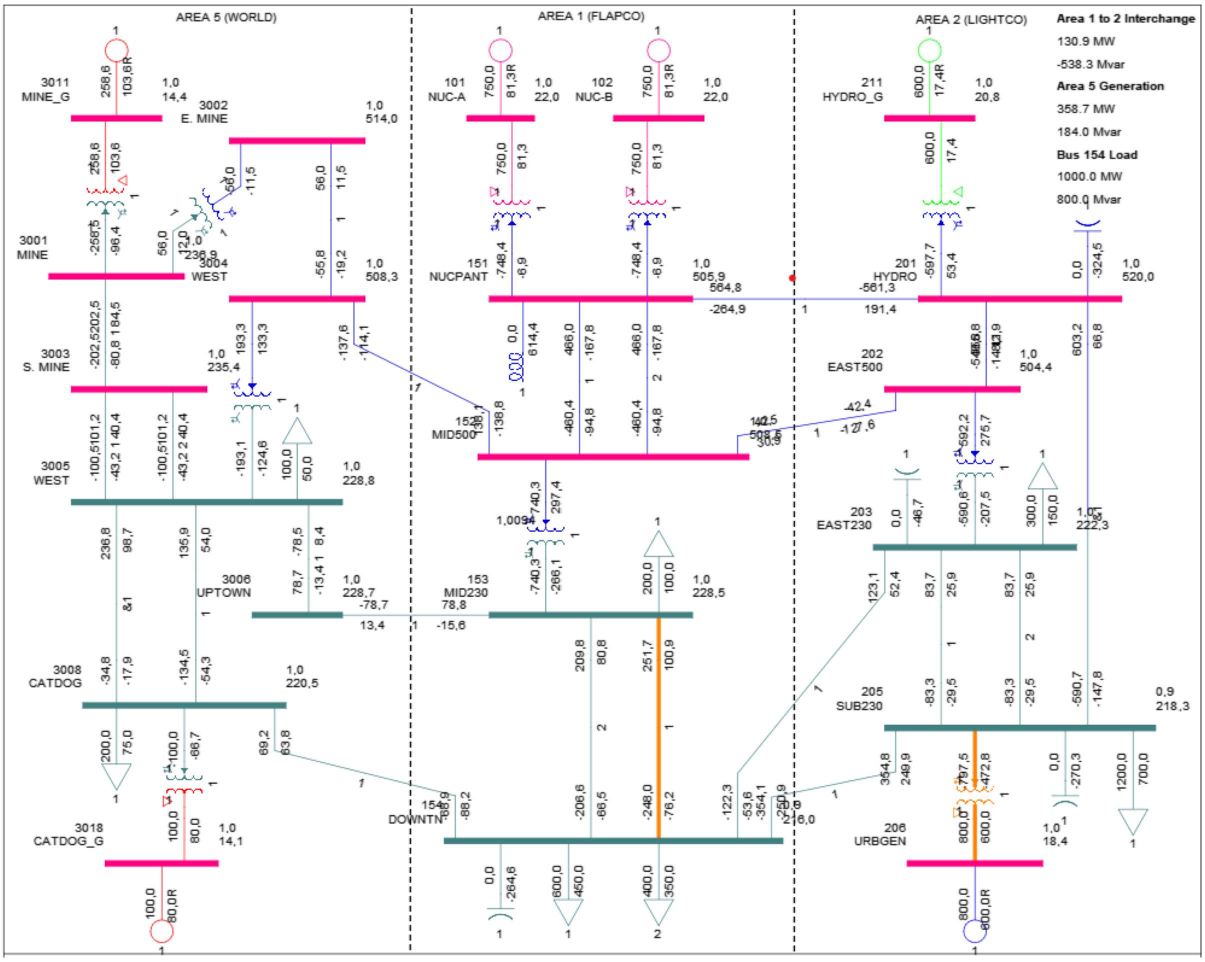


Figure 32: A more detailed one-line diagram of SIEMENS test system called "savnw".

Table 10: Information about the start values at buses for dynamic simulation.

Bus Number	Bus Name	Base kV	Voltage (pu)	Angle (deg)	Normal Vmax (pu)	Normal Vmin (pu)	Emergency Vmax (pu)	Emergency Vmin (pu)
101	NUC-A	21,6	1,02	16,55	1,1	0,9	1,1	0,9
102	NUC-B	21,6	1,02	16,55	1,1	0,9	1,1	0,9
151	NUCPANT	500	1,01	10,89	1,1	0,9	1,1	0,9
152	MID500	500	1,02	-1,12	1,1	0,9	1,1	0,9
153	MID230	230	0,99	-3,24	1,1	0,9	1,1	0,9
154	DOWNTN	230	0,94	-9,88	1,1	0,9	1,1	0,9
201	HYDRO	500	1,04	6,16	1,1	0,9	1,1	0,9
202	EAST500	500	1,01	-1,32	1,1	0,9	1,1	0,9
203	EAST230	230	0,97	-6,92	1,1	0,9	1,1	0,9
204	SUB500	500	0,98	-3,73	1,1	0,9	1,1	0,9
205	SUB230	230	0,95	-9,18	1,1	0,9	1,1	0,9
206	URBGEN	18	1,02	-2,97	1,1	0,9	1,1	0,9
211	HYDRO_G	20	1,04	12,92	1,1	0,9	1,1	0,9
3001	MINE	230	1,03	-1,37	1,1	0,9	1,1	0,9
3002	E. MINE	500	1,03	-1,83	1,1	0,9	1,1	0,9
3003	S. MINE	230	1,02	-2,25	1,1	0,9	1,1	0,9
3004	WEST	500	1,02	-3,43	1,1	0,9	1,1	0,9
3005	WEST	230	0,99	-5,18	1,1	0,9	1,1	0,9
3006	UPTOWN	230	0,99	-3,79	1,1	0,9	1,1	0,9
3007	RURAL	230	0,96	-8,54	1,1	0,9	1,1	0,9
3008	CATDOG	230	0,96	-9,05	1,1	0,9	1,1	0,9
3011	MINE_G	13,8	1,04	0	1,1	0,9	1,1	0,9
3018	CATDOG_G	13,8	1,02	-4,08	1,1	0,9	1,1	0,9

Table 11: Information about start values at generator buses for dynamic simulation.

Bus Number	Bus Name	Mbase (MVA)	Pgen (MW)	Qgen (Mvar)	H, inertia (MJ/MVA)
101	NUC-A	900	750	81	4
102	NUC-B	900	750	81	4
206	URBGEN	1000	800	600	2,5
211	HYDRO_G	725	600	17	5
3011	MINE_G	1000	259	104	3
3018	CATDOG_G	130	100	80	3

C. Results from literature search

Transient stability:

Table 12: Table displaying all the relevant articles found about transient stability.

Title	Algorithm	Year
Estimation of rotor angles of synchronous machines using artificial neural networks and local PMU based quantities	ANN	2007
Real-Time Monitoring of Post-Fault Scenario for Determining Generator Coherency and Transient Stability through ANN	ANN	2017
Wide Area Measurement-based Transient Stability Prediction using Long Short-Term Memory Networks	LSTM	2017
Intelligent Time-Adaptive Transient Stability Assessment System	LSTM	2019
A new method of decision tree based transient stability assessment using hybrid simulation for real-time PMU measurements	DT	2020
Transient stability assessment via decision trees and multivariate adaptive regression splines	DT	2016
Stacked-GRU Based Power System Transient Stability Assessment Method	Gated Recurrent Unit (GRU)	2018
Real-time transient stability prediction of power systems based on the energy of signals obtained from PMUs	MLP, DT, Naïve Bayes classifiers	2020

LFO:

Table 13: Table displaying all the relevant articles found about LFO.

Title	Algorithm	Year
Intelligent Wide Area Monitoring of Power System Oscillatory Dynamics in Real Time	ANN	2017
Wide-area PMU-ANN based monitoring of low frequency oscillations in a wind integrated power system	ANN	2018
Prediction of Electromechanical Oscillatory Parameters in Power Systems Using ANN	ANN	2021

Fault prediction:

Table 14: Table displaying all the relevant articles found about fault prediction.

Title	Algorithm	Year
PMU Analytics for Power Fault Awareness and Prediction	DT and SVR	2019
Support Vector Machine Based Proactive Cascade Prediction in Smart Grid Using Probabilistic Framework	SVM	2014
Augmenting WAMPAC with machine learning tools for early warning and mitigation of blackout events	ANN	2018
Neural Network Based Early Warning System for an Emerging Blackout in Smart Grid Power Networks	ANN	2015

Fault prediction without PMU:

Table 15: Table displaying all the relevant articles found about transient stability without PMU.

Title	Algorithm	Year
Machine Learning to Predict Fault Events in Power Distribution Systems	ANN	2021
Machine Learning based Power Grid Outage Prediction in Response to Extreme Events	LR	2017
Three–Phase Line Overloading Predictive Monitoring Utilizing Artificial Neural Networks	ANN	2017
Improving Power Grid Resilience Through Predictive Outage Estimation	SVM	2018
Data-Based Line Trip Fault Prediction in Power Systems Using LSTM Networks and SVM	LSTM and SVM	2018
Data-Driven Multi-Hidden Markov Model-Based Power Quality Disturbance Prediction that Incorporates Weather Conditions	Multi-Hidden Markov Model	2019
Computerized System for Detection of High Impedance Faults in MV Overhead Distribution Lines	ANN	2016

D. Python code

```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Sun May 23 10:24:18 2021

@author: tobiaskorten
"""

#%% Import data from PSSE (as excel files).

import pandas as pd
import numpy as np

stable_urbgen1 =
pd.read_excel('/Users/tobiaskorten/Documents/Masteroppgave/Python_masteropp
gave/Excel/Stable/stable1_urbgen.xlsx', header=1, names = np.array(['Time',
'Stable 1: URBGEN rotor angles']))
stable_urbgen2 =
pd.read_excel('/Users/tobiaskorten/Documents/Masteroppgave/Python_masteropp
gave/Excel/Stable/stable2_urbgen.xlsx', usecols = 'B', header=1, names =
np.array(['Stable 2: URBGEN rotor angles']))
stable_urbgen3 =
pd.read_excel('/Users/tobiaskorten/Documents/Masteroppgave/Python_masteropp
gave/Excel/Stable/stable3_urbgen.xlsx', usecols = 'B', header=1, names =
np.array(['Stable 3: URBGEN rotor angles']))

stable_mineg1 =
pd.read_excel('/Users/tobiaskorten/Documents/Masteroppgave/Python_masteropp
gave/Excel/Stable/stable1_mine_g.xlsx', usecols = 'B', header=1, names =
np.array(['Stable 1: MINE_G rotor angles']))
stable_mineg2 =
pd.read_excel('/Users/tobiaskorten/Documents/Masteroppgave/Python_masteropp
gave/Excel/Stable/stable2_mine_g.xlsx', usecols = 'B', header=1, names =
np.array(['Stable 2: MINE_G rotor angles']))
stable_mineg3 =
pd.read_excel('/Users/tobiaskorten/Documents/Masteroppgave/Python_masteropp
gave/Excel/Stable/stable3_mine_g.xlsx', usecols = 'B', header=1, names =
np.array(['Stable 3: MINE_G rotor angles']))

unstable_urbgen1 =
pd.read_excel('/Users/tobiaskorten/Documents/Masteroppgave/Python_masteropp
gave/Excel/Unstable/unstable1_urbgen.xlsx', header=1, names =
np.array(['Time', 'Unstable 1: URBGEN rotor angles']))
unstable_urbgen2 =
pd.read_excel('/Users/tobiaskorten/Documents/Masteroppgave/Python_masteropp
gave/Excel/Unstable/unstable2_urbgen.xlsx', usecols = 'B', header=1, names
= np.array(['Unstable 2: URBGEN rotor angles']))
unstable_urbgen3 =
pd.read_excel('/Users/tobiaskorten/Documents/Masteroppgave/Python_masteropp
gave/Excel/Unstable/unstable3_urbgen.xlsx', usecols = 'B', header=1, names
= np.array(['Unstable 3: URBGEN rotor angles']))

unstable_mineg1 =
pd.read_excel('/Users/tobiaskorten/Documents/Masteroppgave/Python_masteropp
```

```
gave/Excel/Unstable/unstable1_mine_g.xlsx', usecols = 'B', header=1, names
= np.array(['Unstable 1: MINE_G rotor angles']))
unstable_mineg2 =
pd.read_excel('/Users/tobiaskorten/Documents/Masteroppgave/Python_masteropp
gave/Excel/Unstable/unstable2_mine_g.xlsx', usecols = 'B', header=1, names
= np.array(['Unstable 2: MINE_G rotor angles']))
unstable_mineg3 =
pd.read_excel('/Users/tobiaskorten/Documents/Masteroppgave/Python_masteropp
gave/Excel/Unstable/unstable3_mine_g.xlsx', usecols = 'B', header=1, names
= np.array(['Unstable 3: MINE_G rotor angles']))


#%% Concatenate the seperate dataframes into two dataframes (stable and
unstable) and drop duplicates.

stable = pd.concat([stable_urbgen1, stable_mineg1, stable_urbgen2,
stable_mineg2, stable_urbgen3, stable_mineg3], axis=1)
unstable = pd.concat([unstable_urbgen1, unstable_mineg1, unstable_urbgen2,
unstable_mineg2, unstable_urbgen3, unstable_mineg3], axis=1)

stable.drop_duplicates(subset ="Time", keep = 'first', inplace = True)
unstable.drop_duplicates(subset ="Time", keep = 'first', inplace = True)


#%% Remove unnecessary samples and append class label column.

stable_new = stable.query('Time >= 1.08 and Time < 1.28')
unstable_new = unstable.query('Time >= 1.08 and Time < 1.28')

stable_new.insert(len(stable_new.columns), 'Stable?', 0)
unstable_new.insert(len(unstable_new.columns), 'Stable?', 1)


#%% Filter out samples such that the timestep is 0.02 seconds and transpose
the dataframes.

stable_values = stable_new[::2]
unstable_values = unstable_new[::2]

stable_valuest = stable_values.transpose()
unstable_valuest = unstable_values.transpose()


#%% Divide the stable samples into the three seperate cases and the two
generators. Filter out the relevant samples for each case.

urbgen_s1 = stable_valuest.iloc[[1], :]
urbgen_s1 = urbgen_s1.iloc[:, 2:-2]
urbgen_s1.set_axis(range(6), axis=1, inplace=True)
urbgen_s1.insert(len(urbgen_s1.columns), 'Stable?', 0)

mineg_s1 = stable_valuest.iloc[[2], :]
mineg_s1 = mineg_s1.iloc[:, 2:-2]
mineg_s1.set_axis(range(6), axis=1, inplace=True)
mineg_s1.insert(len(mineg_s1.columns), 'Stable?', 0)

urbgen_s2 = stable_valuest.iloc[[3], :]
```

```
urbgen_s2 = urbgen_s2.iloc[:, 4:]
urbgen_s2.set_axis(range(6), axis=1, inplace=True)
urbgen_s2.insert(len(urbgen_s2.columns), 'Stable?', 0)

mineg_s2 = stable_valuest.iloc[[4], :]
mineg_s2 = mineg_s2.iloc[:, 4:]
mineg_s2.set_axis(range(6), axis=1, inplace=True)
mineg_s2.insert(len(mineg_s2.columns), 'Stable?', 0)

urbgen_s3 = stable_valuest.iloc[[5], :]
urbgen_s3 = urbgen_s3.iloc[:, :-4]
urbgen_s3.set_axis(range(6), axis=1, inplace=True)
urbgen_s3.insert(len(urbgen_s3.columns), 'Stable?', 0)

mineg_s3 = stable_valuest.iloc[[6], :]
mineg_s3 = mineg_s3.iloc[:, :-4]
mineg_s3.set_axis(range(6), axis=1, inplace=True)
mineg_s3.insert(len(mineg_s3.columns), 'Stable?', 0)


#%% Concatenate the seperate dataframes into one dataframe for stable
values.

stable_last = pd.concat([urbgen_s1, mineg_s1, urbgen_s2, mineg_s2,
urbgen_s3, mineg_s3], axis=0, ignore_index=False, sort=True)


#%% Divide the unstable samples into the three seperate cases and the two
generators. Filter out the relevant samples for each case.

urbgen_us1 = unstable_valuest.iloc[[1], :]
urbgen_us1 = urbgen_us1.iloc[:, 2:-2]
urbgen_us1.set_axis(range(6), axis=1, inplace=True)
urbgen_us1.insert(len(urbgen_us1.columns), 'Stable?', 1)

mineg_us1 = unstable_valuest.iloc[[2], :]
mineg_us1 = mineg_us1.iloc[:, 2:-2]
mineg_us1.set_axis(range(6), axis=1, inplace=True)
mineg_us1.insert(len(mineg_us1.columns), 'Stable?', 1)

urbgen_us2 = unstable_valuest.iloc[[3], :]
urbgen_us2 = urbgen_us2.iloc[:, 4:]
urbgen_us2.set_axis(range(6), axis=1, inplace=True)
urbgen_us2.insert(len(urbgen_us2.columns), 'Stable?', 1)

mineg_us2 = unstable_valuest.iloc[[4], :]
mineg_us2 = mineg_us2.iloc[:, 4:]
mineg_us2.set_axis(range(6), axis=1, inplace=True)
mineg_us2.insert(len(mineg_us2.columns), 'Stable?', 1)

urbgen_us3 = unstable_valuest.iloc[[5], :]
urbgen_us3 = urbgen_us3.iloc[:, :-4]
urbgen_us3.set_axis(range(6), axis=1, inplace=True)
urbgen_us3.insert(len(urbgen_us3.columns), 'Stable?', 1)

mineg_us3 = unstable_valuest.iloc[[6], :]
mineg_us3 = mineg_us3.iloc[:, :-4]
```

```
mineg_us3.set_axis(range(6), axis=1, inplace=True)
mineg_us3.insert(len(mineg_us3.columns), 'Stable?', 1)


#%% Concatenate the seperate dataframes into one dataframe for unstable
values.

unstable_last = pd.concat([urbgen_us1, mineg_us1, urbgen_us2, mineg_us2,
urbgen_us3, mineg_us3], axis=0, ignore_index=False, sort=True)


#%% Concatenate the two dataframes for stable and unstable samples into one
dataframe.

complete = pd.concat([stable_last, unstable_last])


#%% Export the complete dataframe to an excel file.

complete.to_excel('output.xlsx')
```

Norges miljø- og biovitenskapelige universitet
Noregs miljø- og biovitskapelege universitet
Norwegian University of Life Sciences

Postboks 5003
NO-1432 Ås
Norway