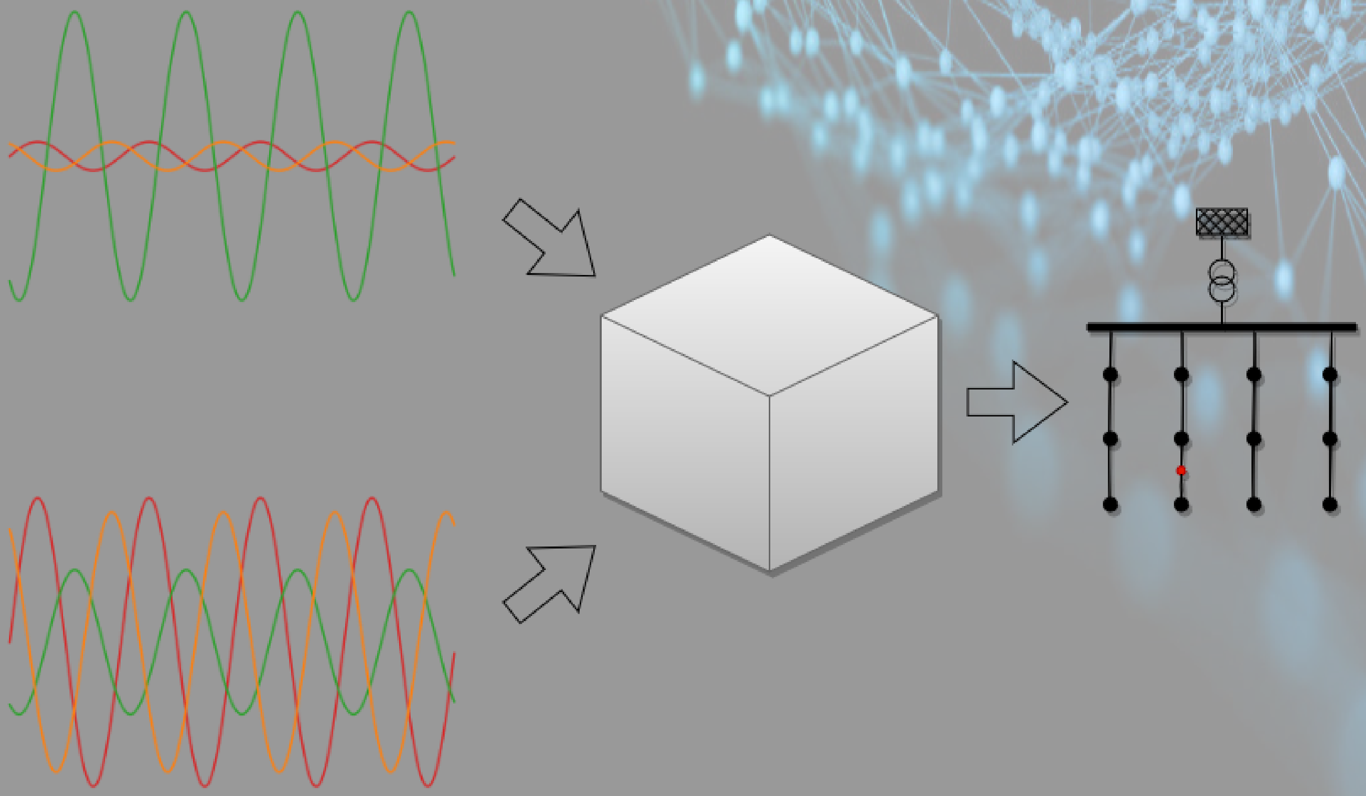
Interpretation of the Machine Learning Fault Classification Model Results Using Explainable Features

Thomas Georgiou

Delft University of Technology



TU Delft

Interpretation of the Machine Learning Fault Classification Model Results Using Explainable Features

by

Thomas Georgiou

to obtain the degree of Master of Science
at the Delft University of Technology,
to be defended publicly on Wednesday September 14, 2022 at 14:15 AM.

Student number:	5395186		
Project duration:	22.11.2021 – 14.09.2022		
Thesis committee:	Prof. dr. Peter Palensky	Chair	IEPG, TU Delft
	Dr. Simon Tindemans	Supervisor	IEPG, TU Delft
	Ir. Tongyou Gu	Supervisor	Qirion, Alliander
	Dr. ir. Mohamad Ghaffarian Niasar	Committee Member	DCE&S, TU Delft

Acknowledgments

This thesis remarks on my final work as part of my MSc degree in Electrical Engineering, bringing up the end of a huge chapter of my life at the Delft University of Technology which I am extremely thankful. It was a challenging however a rewarding journey, to work on a demanding and interesting research project in collaboration with the industry. I am immensely grateful for everything I learnt and the people I learnt from. I would like to use this space to thank the people who contributed into making this thesis possible.

First, I would like to thank my supervisor at the Delft University of Technology, Dr. Simon Tindemans. His constant guidance and experience contributed significantly to my research and his feedback always pushed me to do my best. A special thanks to my supervisors at Qirion/Alliander, Dr.ir. Frans Provoost and Ir. Tongyou Gu for providing me with the opportunity to work on this project. Their expert knowledge in electrical power engineering and their support were crucial to my efforts throughout my thesis and helped me learn a lot. The collaboration with my supervisors was exceptional and all together ensured my research went in the right direction while extending my knowledge about data science and power engineering. I would also like to express my gratitude to Prof. dr. Peter Palensky for being the head of my thesis committee and participating in all the milestones during my research, providing valuable input. Furthermore, I would also like to thank Dr.ir. Mohamad Ghaffarian Niasar for agreeing to be part of my thesis committee and evaluating my work and report. Finally, I would like to thank my family and friends for their support, patience and encouragement during the whole journey.

Thomas Georgiou
Delft, September 2022

Abstract

Electrical faults in the distribution network can lead to interruptions in the power supply of the customers. Therefore penalties are applied to the DSOs if they overcome the benchmark set based on all the DSOs reliability performance. Hence, the fast restoration of the power supply is crucial for the grid operator in order for the operational costs to be decreased. The traditional way of fault locating is performed with the help of so-called Fault-current Passage Indicators. This can be improved by automatic estimation of the fault location using analysis of the current and voltage signals during the fault, which also depends on the accurate fault classification. However, the existence of distortions and instabilities in some of the fault waveforms leads to an unreliable fault loop-impedance/-reactance. As a result, the location of the fault has to be performed in the traditional way which leads to a delay in the restoration of the power supply.

The objective of this thesis is to explore the potential of modern signal processing and machine learning (ML) techniques for the development of an explainable classification scheme of faults in the distribution network. ML explainability is a huge part of the research held for this thesis, as it is very important for the fault analysis department to understand the model and for the grid operators to trust the results of the model. The problem is divided into two parts. The first part concerns the construction of an explainable ML-based classification model that can accurately differentiate between types of stable faults. The second part concerns the identification of suitable criteria for single-phase and multi-phase fault stability. Combined with a fault classifier based on the first part research, this will lead to a classification and location scheme that can incorporate all fault types.

First, a literature study is done. The pre-modeling explainability stage is the most suitable one and based on that, the Short-time Fourier Transform (STFT) is chosen as the signal processing technique since it can lead to more explainable features than other techniques, e.g. the Wavelet Transform (WT). Also, the Support Vector Machines (SVMs) are selected as the supervised learning technique, as it performs well in classification problems and it is not a method as complicated as for example the Artificial Neural Network (ANN). Next, three different feature sets are shaped appropriately from the three-phase fault current and voltage signals, which are the inputs for the SVM classifier model, so then this model can be tuned and evaluated. The model with the symmetrical feature set is selected serving suitably both the performance and explainability requirements. Following that, a set of stability rules are developed to better identify the stable faults that previously were classified as unstable. These rules are tuned based on a set of initially labeled unstable faults. In the end, the ML classification and the stability model combined with section selection rules constitute the final classification scheme. This is tested on hidden data and the results indicate good performance in classifying faults, locating faults and identifying stable faults that initially would be classified as unstable. In particular, the performance of the proposed scheme on that test set has f1-score of 95.1 % exceeding 93.3 % of the current algorithm tested on the same data, errors regarding the fault loop-impedance/reactance are located within the acceptable limits and 4 stable faults correctly identified that previously were seen as unstable.

Contents

1 Introduction 1
1.1 Background 1
1.2 Motivation 2
1.3 Thesis Objectives and Scope 3
1.4 Research Questions 4
1.5 Thesis Structure. 4

2 Literature Review 5
2.1 Introduction to Faults in the Distribution Network 5
2.2 Present Fault Location Process at Alliander 6
2.2.1 Overview of the Fault Location Process 6
2.2.2 Previously Proposed Fault Classification Methodology. 10
2.3 Review of Explainability in Machine Learning. 12
2.4 Review of Classification Methods for Electrical Faults 13
2.4.1 Review of Feature Engineering Techniques. 13
2.4.2 Review of Fault Classification Techniques 14
2.5 Summary 15

3 Theoretical Background 17
3.1 Description of Dataset 17
3.2 Machine Learning. 19
3.2.1 Supervised Machine Learning 19
3.2.2 Support Vector Machine 20
3.3 Metrics for Evaluating the Model's Performance 22
3.3.1 Metrics for Evaluating the Classification Model's Performance. 22
3.3.2 Metrics for Evaluating the Model's Performance on Calculating the Loop-Impedance . . . 23
3.4 Fault Segmentation Algorithm 24
3.4.1 Binary Segmentation Algorithm 24
3.4.2 Segmentation Algorithm Implementations 25
3.4.2.1 Univariate segmentation algorithm 25
3.4.2.2 Multivariate segmentation technique 26
3.5 Summary 27

4 Classification Models with Explainable Features for Stable Faults 29
4.1 Development of Explainable Features for Stable Faults 29
4.1.1 Development of Three Models 29
4.1.2 Features based on the Short-time Fourier Transform Results. 30
4.1.3 Features based on the Symmetrical Components Ratios 32
4.1.4 Statistical Metrics used for the Feature Extraction 33
4.1.5 Features from the Voltage signal 33
4.1.6 Features from the Current signal 34
4.1.7 Features from the combination of Voltage and Current signals. 35
4.2 Model Development 36
4.2.1 Feature Selection. 37
4.2.2 Hyper-parameter tuning 38
4.2.3 Cross-validation 39
4.3 Results 40
4.3.1 Results for the Model 1 with Features in Order 40
4.3.2 Results for the Model 2 with Per-Phase Features 41
4.3.3 Results for the Model 3 with Symmetrical Features. 41
4.3.4 Comparison of the three models 42

4.4 Summary . . . 43

5 Development of Model for Unstable Faults 45

5.1 Model Development . . . 45

5.2 Development of Features for Stability . . . 49

5.2.1 Feature Development . . . 49

5.2.2 Feature Evaluation . . . 51

5.2.3 Stability Rules Model Results . . . 52

5.3 Results of the Entire Model . . . 57

5.3.1 Classification Model Selection . . . 57

5.3.2 Results of the Model for Fault Classification and Fault Location . . . 57

5.3.3 Discussion . . . 60

5.4 Summary . . . 62

6 Conclusion 63

6.1 Research Questions . . . 63

6.2 Future Research . . . 65

A Supplementary Figures 73

B Supplementary Results 77

1

Introduction

1.1. Background

Traditionally electricity networks transfer electrical power from generation to demand in a downstream way, from high voltage (HV) to medium voltage (MV) and in the end to low voltage (LV) and the end consumers. The main feature of the the distribution network was to distribute the power from the HV/MV substation to the end customers in that uniform direction. The increase of distributed generation located in the MV and LV grid, such as the solar parks and rooftops, small-scale wind turbines, electric vehicles has changed the traditional way of production and consumption. Also, the growth of electricity consumption per capita leads the system towards its limits and creates the need for grid expansion. Moreover, the Netherlands is trying to move away from the natural gas as a heating resource which turns the house appliances into electrical, hence increasing more the electricity demand. All these changes in the energy production and consumption in the distribution system, make it more powerful, however more vulnerable. As a result the reliability of the system becomes very essential.

Alliander is the largest distribution system operator (DSO) in the Netherlands with 38% market share and it is responsible for the operation and expansion of energy networks. It is important to offer high power quality and reliability of the electricity supply. A very significant aspect that ensures a high quality and reliability is the fast location of a fault and the subsequent quick restoration of the power supply. In general, a fault in the power system could be characterised an abnormal state in which large currents, even in the kA range, are induced and they are cleared by switching off protective devices, such as circuit breakers. The time needed for the fault to be located and then the time for that section to be isolated from the rest of the network constitutes the interruption time. Therefore, the desire is for the interruption time to be as low as possible, so the customers are out of power supply as little as possible.

In order to reduce the outage time a fast localisation process is required. The current process used by Alliander depends on the type of the fault, as different equations are used for each case. So, if the fault classification model is not accurate, that effectively leads to inaccurate locations and an increased interruption time. If the fault is self-extinguishing, which essentially does not trip the circuit breaker (CB) or an unstable one, a possible fault location cannot be sent to the control room. Currently, Alliander uses an identification and location scheme that compares certain measured properties of the fault (current, voltage, impedance, reactance) with preset thresholds that are set based on past fault instances. The existence of the SASensor which is a substation automation tool that stores the current and voltage waveforms of a fault at the substation, encourages the use of machine learning (ML) -based techniques for the recognition of the fault type. However, since these methods make the model a black-box, techniques that add explainability and interpretability in the results and convert it to a grey-box are required so it can be used in practice.

1.2. Motivation

Figure 1.1 depicts a typical structure of a Dutch MV distribution network. The dutch distribution systems usually have ring architecture, however the normal operation is radial. For that there exist normally open points (NOP) that can connect feeders with a CB or a disconnector. Such points close the CB accordingly in case of maintenance or fault on a cable section, so the power supply can be restored to the network beyond that section with fewer customers affected [1]. When a fault occurs on a section in the distribution network, the CB at the MV busbar opens. So the supply of that feeder stops until the exact location of the faulted cable connecting the units is found and get isolated. Then the power supply is restored. Hence the importance of locating the faulted section is that high. The fault location estimation is determined based on the fault loop-impedance which is compared with the impedance derived from the Vision FaultFinder software, which is a network analysis tool. The loop-impedance calculation depends on the type of the fault though and thus the fault type classification accuracy is essential. A misclassification would result to a wrong loop-impedance value which lead to a not proper estimation of the fault location.

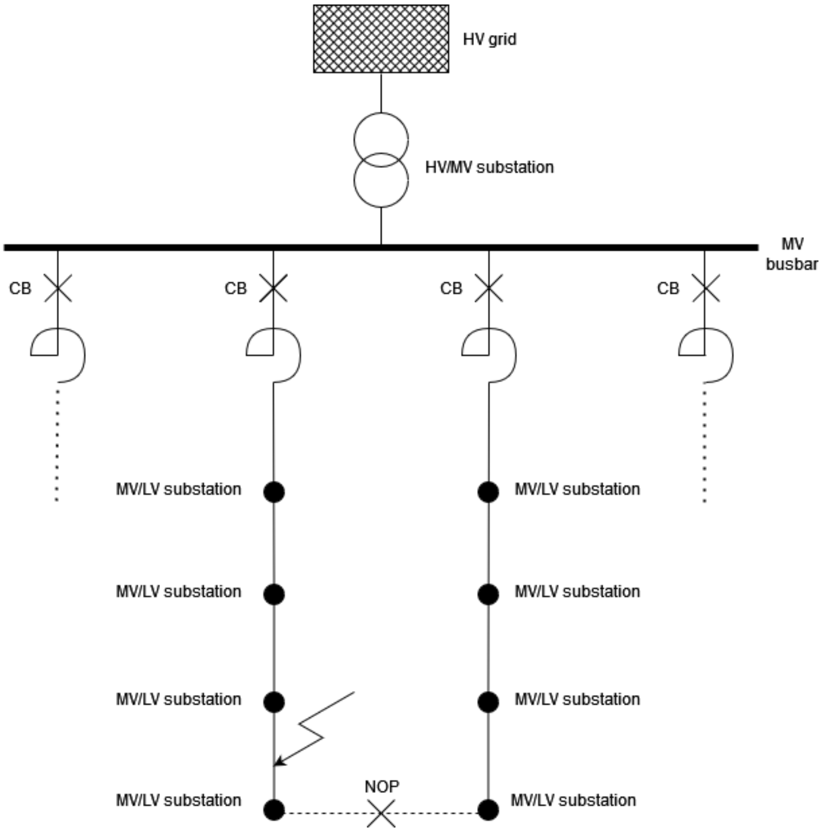


Figure 1.1: Typical structure of a Dutch MV distribution grid [1]

The existing fault classification model of Alliander is quite accurate. However, the option of replacing the current fault classification model with a ML model was chosen based on exploiting the data being stored by Alliander regarding the faults in the system. Alliander uses the SASensor in a number of substations. This system sends every suspect waveform that surpasses the preset thresholds to the control room for further analysis, so the entries that correspond to faults could be useful for the fault classification. This database is only getting larger so developing a ML-based algorithm has also a view to the future use.

The investigation of building such a ML-based classification model was done before by Ranganathan [2], however there are still some weaknesses that make the use in practice still infeasible. The main issue is the need from the fault analysis department to understand the model and the grid operators, the users of the Vision FaultFinder software to trust the results of that model. This ML-based model then is operating as a black-box and this specific application which concerns the fault location and the interruption time after a fault, making the use of such a model impractical.

Moreover, it is required to have a better set of rules regarding the classification of faults as unstable. An unstable fault is not used for fault location, so the fault locating is performed in the traditional way. That inevitably leads to an increased interruption time, compared to a stable fault passing through the automatic procedure. After the fault is located by the traditional fault location process, the recorded data is analysed by experts who manually calculate a fault location. This location is then compared with the actual one. In several cases it appeared that the location given by the unstable fault was the correct one. The definition of unstable faults is given by the thresholds presented by Gu and Provoost [3] which can be described as conservative. Therefore, faults that could be classified as stable ones and could provide a reliable loop-impedance, occasionally are classified as unstable if some of these thresholds are surpassed. So, a refinement of these stability rules is required, so for less eventually stable faults only the traditional and time-consuming procedure of fault location is used. Ranganathan [2], formulated some accurate and better rules in terms of that. However, rules that can be applied to multi-phase faults which are also explainable are needed.

In the model developed that satisfies the explainability and stability issues should as well be added the recognition of self-extinguishing faults. These consist of events of short duration which do not trigger any CB and do not need direct location. However these faults are warnings of a possible real fault. So when a few of these self-extinguishing faults are detected, the Smart Cable Guard system is used to pinpoint the weak point, which can be repaired before the real fault happens. This saves a lot of customer minutes lost. The methods for the preventive location of these faults are explained by Gu and Provoost [4] and they are out of this thesis' scope. The importance of accurately recognize them is that a lot of trouble is saved since a possible loop-impedance calculated from those most probably leads to a non-faulty section.

Overall, the goal is to find ways for the location to be faster, that leads the minimization of the interruption time and subsequently to a more reliable system. The reliability is measured by the metrics Customer Average Interruption Duration Index (CAIDI) for the customer side and System Average Interruption Duration Index (SAIDI) for the system side, based on the time that power is not supplied by the electric grid. The regulator in the Netherlands benchmarks the DSOs according to their quality of performance or q-factor and uses a penalty and reward system [5]. This is an incentive for the DSOs to strive keeping a low customer and system interruption time.

1.3. Thesis Objectives and Scope

The goal of this thesis project is to investigate the potential of an explainable Machine Learning (ML) model for the classification of faults in the distribution network. First, an overview of the methods existing in the literature for inserting explainability to an artificial intelligent (AI) model is held. Then modern classification models are analysed from the literature review, in order to end up to the best combination. The focus is on the feature extraction and the signal processing techniques used for the faults waveform analysis first for the stable and then for the unstable faults.

Therefore, the thesis objective can be distinguished into two sections. One is to to develop an accurate and explainable ML fault classifier algorithm that can identify the type of the stable faults and the other is to formulate suitable rules and extend the classification model so the unstable faults are better identified.

The scope of the study depends on the reliable fault data available. This data represent real faults that sometimes deviate from the ones investigated in ideal circumstances, so these difference should be known. Insights are then made on the performance of the ML classifier on stable and unstable faults, under the boundary of the explainable features. In the next section, the thesis objectives are presented in the form of research questions that are intended to be answered over this thesis project.

1.4. Research Questions

The following are the research questions that are addressed in this thesis. The overarching question is:

How can modern signal processing and machine learning techniques be used to construct an accurate and explainable fault classification scheme in the distribution network?

The sub-questions that are answered during the course of this thesis are:

1. Which explainability approaches exist for a machine learning model and how to evaluate them?
2. What explainable features could be extracted for the fault classification?
3. How can the unstable faults that can provide fault location be better distinguished than the current scheme?
4. What are the methods to evaluate the machine learning model's performance on classifying the faults?
5. What are the methods to determine the best time period for the fault-loop impedance calculation for accurate fault location estimation?

1.5. Thesis Structure

Figure 1.2 summarises the thesis structure.

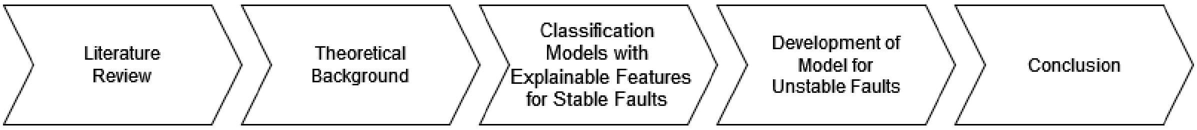

Literature Review
Theoretical Background
Classification Models with Explainable Features for Stable Faults
Development of Model for Unstable Faults
Conclusion

Figure 1.2: Outline of the thesis

First, in chapter 2 the current fault classification and location techniques are presented, as well as a literature survey regarding the explainability in artificial intelligent (AI) models and the fault classification. The different methods were discussed and the relevant choices were identified, so they could provide answers to the research questions. Chapter 3 provides the required theoretical background about the dataset used, the ML and the segmentation techniques that are further used in this project. Next, chapter 4 presents the development of classification models for stable faults which have explainable features. The process of the development based on the feature engineering is done and a then a final selection of model to be used for the stable faults is given. Chapter 5 describes the procedure of developing a classification model for both stable and unstable faults, that in the end extends to the fault location estimation. This is done, after a refinement of the stability rules is held and evaluated by the final loop-impedance/-reactance, similar to the final model's performance evaluation. Finally, in chapter 6 the final conclusions are summarized and the research questions are answered based on the thesis findings. Proposals for future research avenues are formed.

2
Literature Review

This chapter introduces the faults in the distribution system and the explainability in Machine Learning (ML). First, in section 2.1 the fault theory in distribution network is presented, then in section 2.2, the current Alliander fault location system is explained. Section 2.3 shows the explainability techniques available for ML based models and following that section 2.4 reviews the classification methods for a fault in the literature. Finally, section 2.5 sums up the major findings of this chapter.

2.1. Introduction to Faults in the Distribution Network

Fault in a distribution network is called any undesirable deviation from the normal operation conditions and it can cause equipment failure and power outages [6]. Usual cause of such failures are a short circuit that derives from the physical contact between lines after an insulation failure, digging, lightning and a contact with trees or birds [7]. In the Alliander's network there exist mostly underground cables, so the last two causes of failure do not affect it [1].

We can distinguish the faults between symmetrical and unsymmetrical ones. Symmetrical faults are only the three-phase faults, where all the three conductors have a contact with each other and/or the ground. In those faults, the only symmetrical components' sequence existed is the positive one, exactly like at the normal condition. They are usually developed as a result of a previous unsymmetrical fault. They consist of the most severe type of fault and at the same time the rarest type, since they involve all the three phases. In contrast, unsymmetrical faults derived from the contact between a phase conductor and the earth (single-phase to ground fault), two conductors (two-phase fault) and two conductors and the earth (two-phase to ground fault). The single-phase to ground faults are the most frequent type of fault appearing in the system with nearly 70-80% of faults being this type of fault [6].

Moreover, the faults could be distinguished between temporary and permanent ones. Temporary faults are called the self-extinguishing faults that last for a very short-time period due to defects or localized degradation caused by aging in the cable insulation. More precisely, the short self-extinguishing faults last normally for about 2 ms and the long self-extinguishing faults for approximately 10 ms and they get extinguished at the next zero-crossing of the fault current [4]. If the insulation is not healing or getting repaired on time, a series of such faults may lead to a permanent fault and subsequently to a loss of supply. Self-extinguishing fault waveforms can also be characterised by a transient with high frequency that is superposed on steady-state of the signal. The discharge of the faulted phase creates transient behaviour [8]. Figure 2.1 shows the waveform of a single-phase self-extinguishing fault at phase A. Permanent faults last for longer than four cycles and must be switched off externally by a circuit-breaker.

The permanent faults can be further characterised by their stability. Stable faults are called the faults which have current and voltage waveforms that are at a satisfactory point close to the ideal sinusoidal waveform [3]. The definition for the waveform stability here comes based on the Alliander relevant rules and corresponds to sufficiently sinusoidal signals whose results regarding the loop-impedance can be trusted.

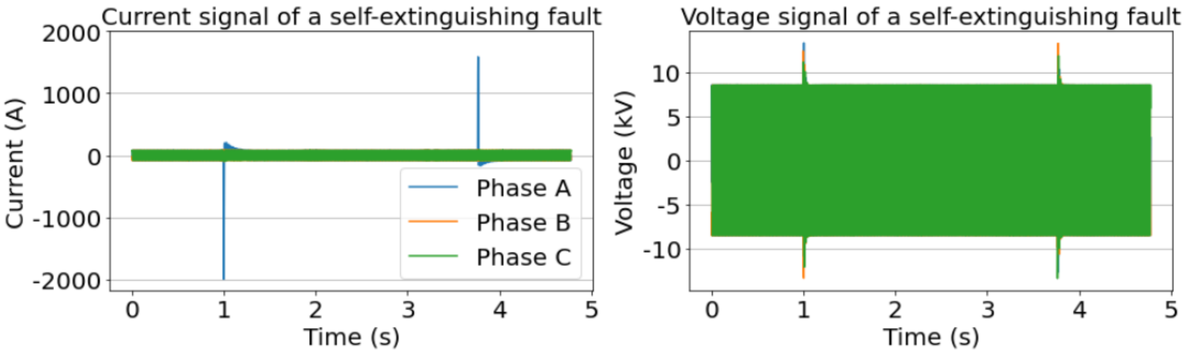


Figure 2.1: Current and voltage waveforms for a self-extinguishing single-phase fault

A formal definition for stable faults has not been instituted yet, so wherever the word stable is used in this project, refers to the definition based on Alliander rules. Unstable faults are the events with waveforms that cannot meet these thresholds, hence they cannot lead to a reliable loop impedance and subsequently to an accurate fault location estimation. In the following chapters these rules are explored too so the stable and unstable sinusoidal waveforms can be better distinguished. Figure 2.2 presents an unstable single-phase fault at phase B.

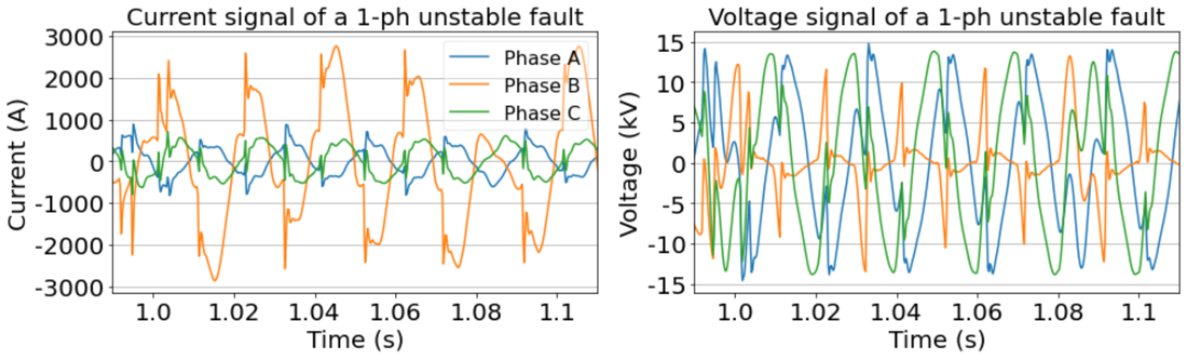


Figure 2.2: Current and voltage waveforms for an unstable single-phase fault

2.2. Present Fault Location Process at Alliander

In this section the present fault location process at Alliander is presented. The discuss then is focused on the fault classification algorithm used, as well as the one previously proposed by Ranganathan [2].

2.2.1. Overview of the Fault Location Process

The current fault location process is shown in 2.3. First, there is the digital fault recording (DFR) of the three-phase voltage and currents through the SASensor at the substation. SASensor is a substation automation platform that measures and monitors the electricity flow within the substation at a high sampling rate of 28kHz. When these measurements exceed the predefined limits, the SASensor stores these waveforms as DFR at a sampling rate of 4kHz which corresponds with 80 samples per period [9].

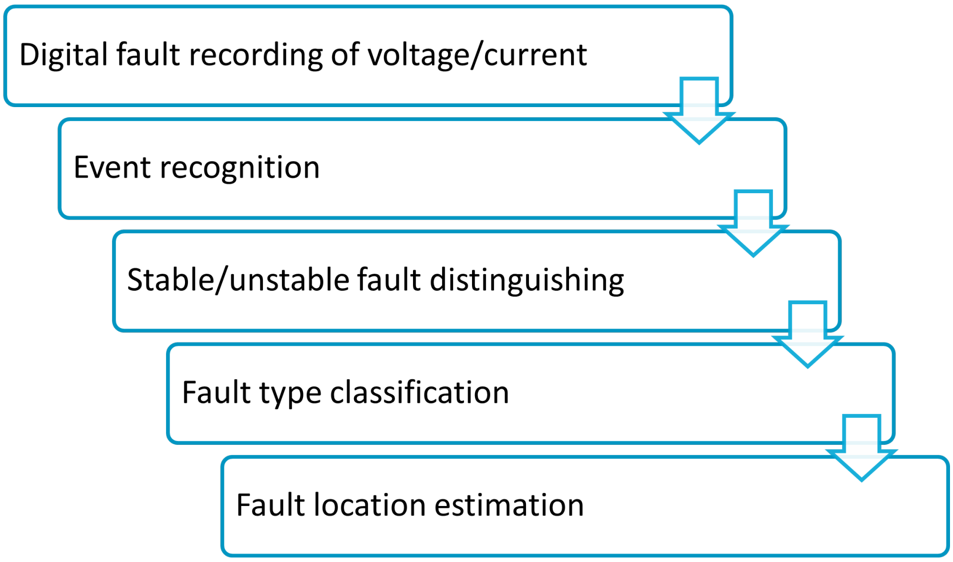

Digital fault recording of voltage/current
Event recognition
Stable/unstable fault distinguishing
Fault type classification
Fault location estimation

Figure 2.3: Current fault location process stages

The event recognition stage is the following one. Once the DFR is created, the corresponding event needs to be recognized. The events of main importance are the faults however due to exceeding the current or voltage thresholds, sometimes the event recorded may correspond to something else, such as inrush current or temporal overcurrent. Then, the faults are further categorized in permanent and self-extinguishing faults. Permanent faults trip the relay and the circuit breaker (CB) opens, so there is an immediate need to quickly isolate that faulty section of the network. These faults are the ones that are being further examined for their location to be found, since there has to be a fast restoration. Self-extinguishing faults initially do not trip the relay however they could be seen as precursor to permanent faults, as the isolation of the cable may be failed. The importance of distinguishing this type of faults compared to the inrush current event, lies on preventing a following developed fault. Therefore, using techniques as Gu and Provoost [4], the location of a self-extinguishing fault can be found, so the section could be repaired before a permanent fault comes up. The location of this type of faults is at the scale of days, serving like a warning for the control room and thus it is not in the scope of this project.

The type of the fault for each period is being identified based on the thresholds visible on the table 2.1 as presented by Gu and Provoost [3]. These thresholds concern the loop-impedance and -reactance and the sequence current and voltage. In particular, Z is the loop-impedance, X the loop-reactance, I_1 the positive sequence current, I_2 the negative sequence current, I_0 the zero sequence current, U_0 the zero sequence voltage, $I_{0min,1}$ a minimum zero sequence current threshold, I_{fp} the current of the fault phase and I_{op} the current of other phases. Therefore, the set of thresholds hold determines the type of fault for each period, before the stability is checked and the final fault type is classified. The investigation to see which set of thresholds is valid begins from the most serious type (three-phase) towards the less serious type (single-phase). That way if a certain set holds, the corresponding type is the final answer and the investigation halts. On the other hand, if no set can be fulfilled it means that the event under investigation is not a fault. The threshold values of the loop-reactance derived based on the assumption that the length of an MV feeder is less than $20km$ and that the positive sequence reactance of an MV cable is around $0.1\Omega/km$. The loop-impedance threshold values ensures that there is a fault and not just a high-resistive load. For the single-phase fault the zero sequence current and voltage are preferred since they are predominant in that case.

Table 2.1: Criteria for fault type classification [3]

Signal	**Fault type**			
	Single-phase	Two-phase	Two-phase to ground	Three-phase
Loop impedance [Ω]	-	$\|Z\| \leq 15$	$\|Z\| \leq 15$	$\|Z\| \leq 7.5$
Loop reactance [Ω]	-	$X \leq 6$	$X \leq 6$	$X \leq 2$
Current sequence ratio	-	$\|I_2\|/\|I_1\| \geq 0.5$	$\|I_2\|/\|I_1\| \geq 0.5$	$\|I_2\|/\|I_1\| \leq 0.1$
Zero-sequence current [A]	$\max I_0 > I_{0min,1}$ $\max I_{fp} / \max I_0 \geq 0.5$ $\max I_{op} / \max I_0 < 0.5$	$I_0 \leq 10$	$I_0 > 10$	-
Zero-sequence voltage [V]	$\max U_0 > 1$	-	-	-

Next stage is the distinction between stable and unstable faults. Stable faults have a current and voltage waveform which is close to an ideal waveform of 50 Hz. These are the faults that can be automatically classified since they provide a reliable loop-impedance. Passing through the fault classifier it means that the classification and subsequently the location of the fault is done at the quickest existed way. On the other hand, unstable faults have not a regular waveform and normally they have no reliable loop-impedance available. So, the type and the location cannot be sent automatically to the control center for further isolation of the suspect fault section. Nevertheless, some unstable faults based on the current Alliander classification scheme can provide a reliable fault location. These kind of unstable faults are judged as almost stable ones after visual observation later on by experts who then calculate the fault location. Table 2.2 presents the rules used by Alliander to distnguish between stable and unstable fault periods as shown by Gu and Provoost [3]. There, MAX represents the maximum value of the waveform for the period explored, RMS is the root mean square (RMS) of the waveform at that period, $MaxDelta$ derives from the maximum value of the difference between sequential data points and N_s is the number of samples per period. If these rules are valid for the relevant fault type for a specific period, then this period is stable.

Table 2.2: Criteria for stable and unstable waveform distinguishing [3]

Signal	**Criteria**	**Fault type**		
		Single-phase	Two-phase	Three-phase
phase voltage	$MAX > 0$	✓	✓	✓
	$RMS/MAX > 0.6$	✓	✓	✓
	$MaxDelta/MAX < 8\pi/N_s$	✓	✓	✓
phase current	$MAX > 0$	✓	✓	✓
	$RMS/MAX > 0.6$	✓	✓	✓
	$MaxDelta/MAX < 4\pi/N_s$	✓	✓	✓
zero-sequence voltage	$MAX > 0$	✓		
	$RMS/MAX > 0.6$	✓		
	$MaxDelta/MAX < 8\pi/N_s$	✓		
zero-sequence current	$MAX > 0$	✓		
	$RMS/MAX > 0.6$	✓		
	$MaxDelta/MAX < 4\pi/N_s$	✓		

After that the final classification of each fault signal is held. As per Gu and Provoost [3], a stable fault with two consecutive periods of the same fault type can be classified as a fault of that type. If more than one types of stable faults exist in an entry, then the most serious is the final type. Three-phase fault is the most serious one, then the two-phase fault and in the end the single-phase fault. If there are no two consecutive periods like that, then the whole event is classified as unstable fault and hence a location cannot be found directly based on that. Input of this stage are the signals with fault type and stability labels for each period, whereas output is the final type of the faults which is needed for the next step of the fault location estimation.

The last step is the fault location estimation. The input of this stage is the fault type which is vital, since for each type of fault a different equation is being used for the calculation of the corresponding loop-impedance or -reactance. These equations can be seen in the table 2.3, where U_0, U_1, U_2 are the zero, positive and negative sequence voltage, I_1, I_2 are the positive and negative sequence current, X_1 is the positive sequence reactance and Z_1, Z_0 are the positive and zero sequence impedance. Basically, the equations derive from the sequence circuits analysis for each type of fault. For the three- and two-phase faults, only the positive sequence impedance X_1 is needed and since it does not depend on the earthing properties, it can provide more accurate results. More precisely, since the error margin in Ohms for multi-phase faults is 0.05Ω and given that for multi-phase faults the correspondence $0.1\Omega/km$ holds, the accepted error in distance is $500m$. On the other hand, for the single-phase fault the zero sequence impedance cannot be neglected, so the absolute value of $2*Z_1+Z_0$ is used. That inserts more uncertainty, as the cable type and diameter affects the resistance and the earthing properties the zero sequence components and hence the error margin is larger for the single-phase faults location. In particular, the error margin in Ohms for single-phase faults is 1Ω and an acceptable value that could be taken for the impedance per kilometer is around $0.4-0.7\Omega/km$ depending on the cable type and diameter. Hence, the accepted error in distance for single-phase faults could be around $1.5km$. Figure A.1 in Appendix A shows the values of reactance/km and impedance/km for three-phase and single-phase fault respectively in different distances from the substation. After calculating the loop-impedance or -reactance, this is being compared to the corresponding value derived from the Vision FaultFinder and thus a possible fault location could be found. This location is sent as an indication to the control room, so the search for the faulty section is more precise. Subsequently, the accuracy of the fault location estimation helps the quicker restoration.

Table 2.3: Loop impedance/reactance equations for locating the fault [1]

Fault type	**Reactance/Impedance from the substation to the location of the fault**
Three-phase	$X_1 = Im\left(\frac{U_1}{I_1}\right)$
Two-phase	$X_1 = Im\left(\frac{U_1-U_2}{I_1-I_2}\right)$
Single-phase	$\lvert 2*Z_1+Z_0\rvert = \left\lvert\frac{U_1+U_2+U_0}{I_2}\right\rvert$

Figure 2.4 presents a more detailed diagram of the whole restoration procedure. The previous explained stages cover the whole procedure except for the restoration. As said, sending a possible fault location to the control room, helps spotting the exact location of the fault quicker, so subsequently to lower interruption time and faster outage restoration. Local sensors are used at the section, like the Smart Cable Guard (SCG), for the exact location to be found, thus without an indication for the fault location, the restoration becomes more time-consuming.

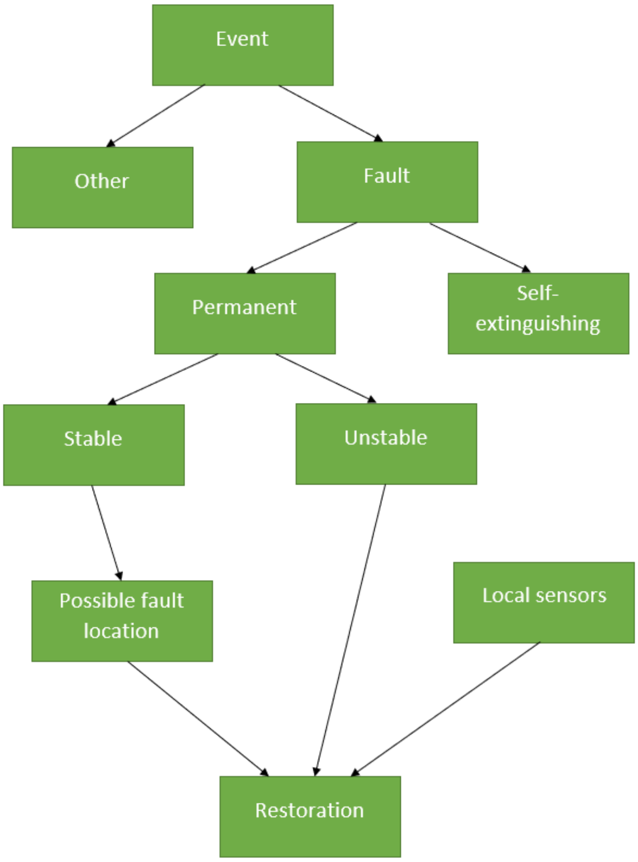


Figure 2.4: Current fault restoration process diagram

2.2.2. Previously Proposed Fault Classification Methodology

In the context of a graduation project, Ranganathan proposed a different fault classification methodology [2]. First of all, the techniques being used in that project are quite different to the ones used in the Alliander's current process. The main differences are derived by the use of Machine Learning techniques for the fault classification algorithm and sliding window Fourier Transform (FT) for formulation of new stability rules.

First, the stable and unstable single-phase faults are distinguished by different criteria, which derive from the features shown in table 2.4. These features are extracted by short-time FT (STFT) and the F_{ratio} which equals:

$$F_{ratio} = \frac{\alpha 0_{fault}}{\alpha 0_{ref}} \tag{2.1}$$

Basically, this reflects to the ratio of fundamental frequency component's magnitude $\alpha 0_{fault}$ of the normalized zero sequence current derivative to the fundamental frequency component's magnitude $\alpha 0_{ref}$ of an ideal reference current per one period. For the former, the derivative corresponds to the point-point difference between consecutive sample points of a signal and then this signal is normalized to unit energy. For the latter an ideal sinusoidal is used only consisted by the fundamental frequency component and unit energy. Since $\alpha 0_{ref}$ is always equal or greater than $\alpha 0_{fault}$, the values of the F_{ratio} are restricted by the interval [0,1] and an ideally stable fault period should correspond to $F_{ratio} = 1$. So, the thresholds of these features' values in order to determine if a fault is stable or not, are calculated by cross-validation on the whole dataset. These values can be seen in table 2.4. The business rules developed by Ranganathan for the stability check involve a 5-point scoring system. Each threshold that holds, provide a score 1. Therefore, if the total score is 5 the fault is definitely a stable fault. On the other hand if the score is 0 the fault is definitely an unstable fault. If the score is 4 and both the two top thresholds hold, the fault is stable and if the score is 1 and both the two top thresholds do not hold, the fault is unstable. The rest of the cases consist of unstable faults that need manual inspection for the calculation of their fault location. These rules regard only the cases of single-phase faults, whereas the single-phase self-extinguishing fault is consider an extreme case of unstable fault. So, the scores 0 and 1 with the previously mentioned condition, lead to a self-extinguishing fault recognition which does not require a direct fault location estimation. This technique is applied only in single-phase faults, so in this thesis a technique that can be applied to multi-phase faults is investigated and developed in chapter 5.

Table 2.4: Proposed stability thresholds by Ranganathan [2]

Feature thresholds	**Description**
$f\_max > 0.8$	Maximum value of F_{ratio} over the entire period of the fault
$f\_stdev > 0.123$	Standard deviation of F_{ratio} over the entire period of the fault
$f\_time > 0.015$	Fraction of time that F_{ratio} is greater than a threshold over the entire period of the fault
$f\_periods > 2.5$	Number of periods that F_{ratio} is greater than a preset threshold over the entire period of the fault
$f\_mean > 0.011$	Mean value of F_{ratio} when is greater than a threshold over the entire period of the fault

Second, for the fault type classification, a ML-based model is being used. The features selected for that model have been extracted from the wavelet transform (WT) and the waveform, using segmentation techniques, to isolate the faulted part. The feature list with their description is shown in the table 2.5, while the

Table 2.5: Final feature set used in the proposed fault classification model by Ranganathan [2]

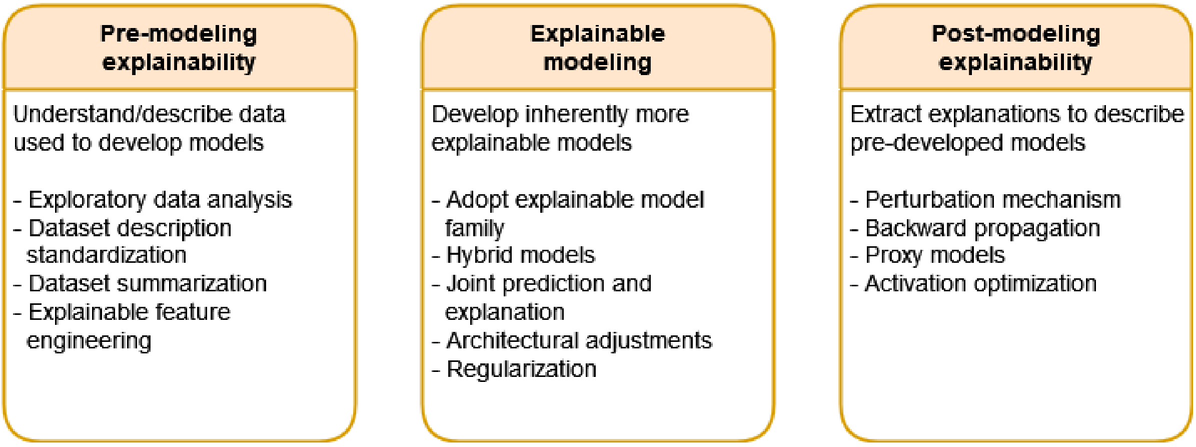
Feature	**Description**
Irms_comparison	Metric for the relationship between the RMS of the phase currents
Irms_2, Irms_3	The smallest and second-smallest RMS values of the phase currents
V0_regular_std, I0_regular_std	Standard deviation of the zero-sequence voltage and current of the fault waveform
i_l1_std2	Middle value of the standard deviation of the detail coefficients of the fault current waveform in the 1-2 kHz bandwidth
i_l1_e3	Smallest value of the energy of the detail coefficients of the fault current waveform in the 1-2 kHz bandwidth
i_l2_e2	Middle value of the energy of the detail coefficients of the fault current waveform in the 0.5-1 kHz bandwidth

hyper-parameters selected for the model are $C = 100$, $\gamma = 0.01$ which were derived by cross-validation. So, here the use of FT per-period is totally abandoned.

Although this whole new method incorporating both the stability rules and the fault classification brings quite accurate results, there are some issues that need to be solved before it could be used as a real application. First, the features used need to be more explainable, so there is a more prominent understanding of the procedure and subsequently it could be trusted by the users. Second, the stability rules concern only the single-phase faults, which is not the only case. It is important to form stability rules for the multi-phase faults, since a waveform can incorporate more than one fault types. Both issues are taken into account for the development of a suitable model in this thesis. The methods followed in order to be resolved the explainability and the stability are further explained in chapter 4 and 5, respectively.

2.3. Review of Explainability in Machine Learning

The introduction of ML models has affected the understanding of how a certain algorithm works, turning the models from white-boxes into black-boxes. In some circumstances, applications do work with the black-box, using the potential of such a model to its fullest [10]. These circumstances are for instance, when the interpretability has no significant impact financial- or social-wise, when there is adequate practical experience with problems being solved by the model, i.e. optical character recognition and when there is a risk of system manipulation by human or program that interpret how the model is making decisions, i.e. bank credit scoring. However, there are quite some applications that need to have a more explainable ML model, a grey-box model [11]. In this section, the levels of explainablity and interpretability are discussed.



Pre-modeling explainability

Understand/describe data used to develop models

- Exploratory data analysis
- Dataset description standardization
- Dataset summarization
- Explainable feature engineering

Explainable modeling

Develop inherently more explainable models

- Adopt explainable model family
- Hybrid models
- Joint prediction and explanation
- Architectural adjustments
- Regularization

Post-modeling explainability

Extract explanations to describe pre-developed models

- Perturbation mechanism
- Backward propagation
- Proxy models
- Activation optimization

Figure 2.5: The three stages of AI explainability: Pre-modeling explainability, explainable modeling and post-modeling explainability [12]

There are three main levels of distinguishing the explainability of a model [12] which are described in figure 2.5. The pre-modeling explainability (feature engineering), explainable model (hybrid model) or post-explainable. which again can be distinguished in model-agnostic (black-box) and model-specific (model related) explainability. Pre-modeling explainability concerns the feature engineering, so the extraction of features that are explainable [13]. The explainable model concerns the ML model itself, since there are ML models which are a priori explainable and interpretable. Those are the linear regression, logistic regression, decision trees (DTs), RuleFit, Naive Bayes and k-Nearest Neighbours (k-NN) [10]. More precisely, using one of this model there is always a way to move from the result back to the model and find out why this was the result. Also, based on those, a lot of hybrid models have been created accompanied by one of those with a non-explainable model i.e. Artificial Neural Network (ANN), Support Vector Machine (SVM), etc. The third category of post-explainable modeling concerns only the way the results can be interpreted. Again here, there is a distinction between the model-agnostic and the model-specific methods and between the local and the global methods [14], [15]. Model-agnostic methods are used as a black-box, without interfering the existing model, whereas model-specific models are model related and have an internal structure. Global methods concern the general trend of a model to reach its results, for example where it gives more importance, while local methods concern specific results derivation. Some global model-agnostic methods are the Partial Dependence Plot (PDP), the Accumulated Local Effects (ALE) and the Global Surrogate. Some local model-agnostic methods are the Individual Conditional Expectation (ICE), the Local Surrogate (LIME) and the Counterfactual Explanations.

Regarding the evaluation of the interpretability or explainability of a model there is not a clear path on how to measure that. However according to Doshi-Velez and Kim [16], there can be three different categories on that. The application level evaluation (real task), the human level evaluation (simple task) and the function level evaluation (proxy task). For the application level evaluation, the model is assessed on the real application with real data by a domain expert. For the human level evaluation the evaluation is carried out not by domain experts, but by laypersons. This is mostly used when the general notion of the quality is needed, allowing for both bigger subject pool and less expenses. On the other hand, function level evaluation does not require humans, which requires class of model that have been already validated as interpretable. So, for example, DT could be used as proxy, for the explanation of another model.

For this thesis project the explainability stage preferred is the pre-modeling one, so more precisely to use explainable feature engineering. This choice seems the proper one, since a more simplistic and explainable model would work in expense of the prediction accuracy. Furthermore, even though the techniques developed for the post-modeling explainability are a lot and they are coming from broad range of methodologies, they still seem inconsistent in their results [17]. Different approaches lead to different conclusions, then for instance if one chooses the technique based on the results that fit him/her, the explainable application will be biased. Regarding the evaluation of the interpretability the most appropriate technique seems to be the application level evaluation from experts, since they could be suitable judges of that because they are going to be either way users of that model. Therefore, the goal is to develop features whose values are going to be sufficiently understood by the users and would justify the results.

2.4. Review of Classification Methods for Electrical Faults

As mentioned previously, in order to find the location of a fault in a power system, first the type of the fault needs to be identified. Since this location is a function of the fault loop impedance and this is dependent on the fault type, the accurate fault classification is vital. More precisely, for each three types of faults, single-phase, two-phase or three-phase fault, a different equation is used for the loop impedance calculation as shown in table 2.2. So, in this section a literature review is presented regarding the ML models being used for the fault classification, as well as the feature extraction techniques. Since the features have to be explainable, there is a focus on reviewing models with such features.

2.4.1. Review of Feature Engineering Techniques

For this thesis project it is significant to overview feature engineering techniques, since this is the epicenter of the explainability of the whole model. The first impression is that the frequency analysis which consists of discrete WT (DWT) and FT, is the predominant signal processing technique. In papers which are using the discrete DWT as a signal processing technique, it is observed a wide range of normalized energy being used for the detail or approximation coefficients. For instance, Magaula et al [18] use normalized energy of level-7 detail coefficients, Livani et al [19], [20] use normalized energy of level-1 and level-2 detail coefficients and Abdelayed et al [21] use normalized energy of approximation and level-1 to level-4 detail coefficients. In papers where FT and sliding window FT are used to analyse the data, the magnitudes of the fundamental frequency components are applied and/or some other frequencies, either harmonics or simply a summation of a frequency range. In particular, Lout et al [22] use the DC component, the fundamental frequency component, the summation of the components over the frequency range 100-550 Hz and 600-1050 Hz, while Aslan et al [23] use the DC component, the 25 Hz component, the fundamental frequency component, the summation of the components over the frequency range 75-200 Hz, 225-500 Hz and 550-1100 Hz. Moreover, Han et al [24] use Short-time FT (STFT) for the fundamental frequency component and the summation of the harmonic components, whereas Ben Hessine et al [25] and Gayathri et al [26] use the fundamental frequency component. When temporal analysis is used for the data, mostly the RMS value of the current/voltage and their harmonics are selected such as in [27], [28] or simply some sampling values around the fault time like in [29].

Another noticeable characteristic in the literature is the signals used as the input of the models. While in a lot of papers only the three-phase voltage and current signals are preferred, in some others the symmetrical components are also shaped and used, providing a phase decoupling to the model, either the zero sequence components [2], [25], [26], [30] or as ratios compared to the positive sequence component [3], [31].

Lastly, the most common element in these techniques for feature engineering is the time period used for the extraction of features useful for the fault type. All these models are using a time period around the fault time deception, either a bit before and a bit after the fault occurs. However, this could lead into mistakes, since real faults often evolve in time and become more severe, due to isolation defections. So, for example, a fault can start as single-phase fault and then can develop into a two-phase fault before it switches off as depicted in figure 2.6. Sometimes it can even further develop into a three-phase fault. So, all these algorithms would focus on the initial fault time period, which would result in a wrong fault type identification and subsequently location. Therefore, it is better to detect and use for feature extraction the time period of the last developed stable fault type as supported by Gu and Provoost [3] and Ranganathan [2].

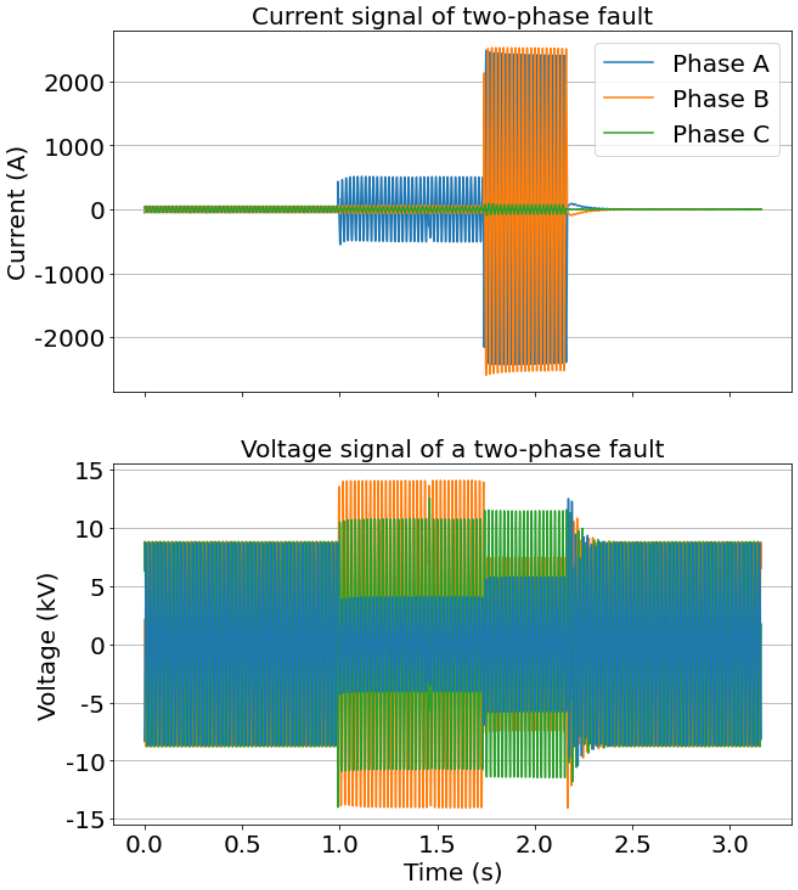


Figure 2.6: Example of a real fault that starts as a single-phase fault and it further develops into a two-phase fault

2.4.2. Review of Fault Classification Techniques

The main importance for this project concerns the fault classification, so it is essential to select an appropriate model for that. It can be observed that the most popular methods proposed for fault classification models are ML-based methods that could exploit the information obtained by the given data. In particular, authors in [18], [19], [20], [32], [33] prefer SVM since it can provide high accuracy reaching the optimal solution for the marginal boundaries among the different classes, whereas authors in [22], [23], [26], [29] choose ANN as it can lead to high performance and robustness. Also, authors in [21], [28] pick DT-based model which has a more straightforward structure and can have quicker training time than ANN, it performs worse than the other ML techniques though. Not all fault classification models are based on ML techniques and instead of that, the features developed for instance from STFT [24], [34], [35] or DWT [29], [30], [36], are extracting directly rules based on certain thresholds regarding the fault classification.

Since the desired techniques used for this project are ML-based, due to the willingness of taking antvantage of the relevant data stored, the selection must be done around SVMs, ANNs and DTs. Additionally, explainability and interpretabilty is even more significant and the model can affect that. An ANN-based model is basically a black-box that offers high accuracy in expense of explainability. On the other hand, DT is a high interpretable model, however due to that, lacks in the accuracy compared to the others. The decision is to use SVM for the fault classification model, since it can provide a more balanced interpretability and accuracy compared to the rest. Therefore, the model developed in this thesis can have as a basis model the one developed from Ranganathan [2], as also the data are similar.

2.5. Summary

First, in this chapter the meaning of a fault in the distribution system was introduced. Subsequently, the current fault location and restoration procedure was explained, with focus on the current and proposed fault classification techniques. After reviewing what explainability in ML is, the development of explainable features was set as the primary source of explainability. Because of that, STFT was selected as the signal processing technique mostly used, as it can offer straightforward results in expense though of performance power compared to WT. At the same context, taking account the trade-off between explainability and performance, SVM is the emerging method to be used in this project as the ML model. Finally, based on the literature review presented in this chapter, some research questions could already have answers.

Which explainability approaches exist for a machine learning model and how to evaluate them? There are three levels of distinguishing the explainability of a model. Pre-modeling explainability (feature engineering), explainable model (e.g. linear regression, DT, hybrid model) or post-explainable, which again can be distinguished in model-agnostic (black-box) and model-specific (model related) explainability. Since it is more important for Alliander what each feature represents, the study will be focused on the pre-modeling explainability and the feature engineering. The evaluation of explainability/interpretability can be done by experts as an application level evaluation or by using a DT as a proxy in a function level evaluation.

What explainable features could be extracted for the fault classification? The less explainable features are predominantly derived by the WT, which according to the literature is being used for the feature extraction for the fault classification in transmission and distribution networks. Also various other signal processing techniques for feature extraction are being used based on the literature study done, such as FT, STFT or even temporal analysis. The most prominent argument against the WT is that the resulted detailed coefficient cannot provide precise frequency or voltage/current information, thus WT can be characterized as less explainable and interpretable compared to STFT. Although STFT or Windowed FT has poorer time-frequency resolution, within each window it can provide precise frequency content information. Since in this project the explainability in the model is vital, it can be used for any transient frequency features needed, instead of WT. Additionally, features based on the fundamental frequency component and the phase angle, such as symmetrical components, impedances, can be extracted using STFT.

3 Theoretical Background

This chapter provides background information needed regarding the developement of the models in the following chapters of this thesis. First, in section 3.1 the kind and the way the data is used in the models is described, in section 3.2 the definition of Machine Learning (ML) is presented with more focus on Support Vector Machines (SVMs). Section 3.3 reflect on the suitable metrics for evaluating the performance of a model on classifying and calculating the loop-impedance respectively. Section 3.4 explores the basis of the segmentation algorithm used for the models developed in the thesis and the different versions of those. Finally, section 3.5 summarizes the key points of the work done in this chapter.

3.1. Description of Dataset

The dataset available by Alliander and the one was used in this thesis concern digital fault recordings (DFRs) of three-phase current and voltage waveforms which reflect to a fault detected in the distribution network. There are different types of fault included in the dataset. In particular, there are single-phase faults both permanent and self-extinguishing ones, two-phase permanent faults, two-phase to ground faults, three-phase permanent faults and unstable faults. The two-phase and two-phase to ground faults are all called with one single label of two-phase faults since they have minimal differences and the two-phase to ground labels correspond to only 6 recordings. The class of unstable faults corresponds mostly to unstable single-phase faults (only 2 unstable two-phase). Table 3.1 sums up the number of samples that correspond to each different fault type. These labels reflect on the real class of each fault provided by Alliander.

Table 3.1: Summary of the number of samples existed in the dataset for each class of fault

True fault type	Number of samples
Single-phase permanent	128
Two-phase	109
Three-phase	94
Single-phase self-extinguishing	432
Unstable	120

First, the result from the Alliander's current model is available which is not for all the types 100% accurate. Then for some faults a manual inspection of their waveform is done by experts afterwards, which leads to the corrected real type of fault which is the one stated above. So, for each recording there exist two labels regarding the fault type. One that corresponds to the output of Alliander's current classification algorithm and one that corresponds to the real type after the manual inspection. For the faults that have no information about the manual inspection, the Alliander's classification model result is considered the real one and it is assumed to be the same as after the manual inspection. Therefore, these are the labels used in order to train correctly the models in chapter 4 and the new proposed model in chapter 5. However also the Alliander's current model results are used for the data recordings that belong to the test set in chapter 5 so to compare the new model with. Most of the misclassifications of the current Alliander's algorithm concern the unstable faults, since the stability rules are a bit conservative. Some other misclassifications come from the self-extinguishing faults, because a series of restriking self-extinguishing faults can lead to a permanent fault which they are in some occasions wrongfully classified as such. Regarding the misclassification of different types of stable permanent faults, there only a few in the data set. They concern 8 faults labeled as single-phase permanent faults, which in reality correspond to 2 two-phase and 6 self-extinguishing faults. This is a result of an old Vision version used. For the multi-phase faults the current classification model is working perfectly.

It is worth mentioning that in chapter 5, a different dataset is used for training and evaluating the ML classification model. The segmentation technique is changing there, then each data recording could result to one or two segmented parts. So, a manual labeling based on the real labels is being done for the recordings that will result in two segments. That means at least one of the segments it should be the same as the real label and the other is assigned manually. However, since each fault sample leads to a single value for its type, the final results for the entire proposed model in that chapter, are evaluated based on their true fault type values.

For the first modeling part of the thesis, the development of classification model for stable faults presented in chapter 4, only the stable permanent faults are considered, hence the single-phase, two-phase and three-phase faults. For the second modeling part of the thesis, the development of classification model that incorporates also the unstable faults and discussed in chapter 5, the whole data set is used. As said above, the segmentation technique is changing there resulting up to two segmented parts, thus a manual labeling is being done based on the real labels. Although the segmentation techniques between these two parts are be-

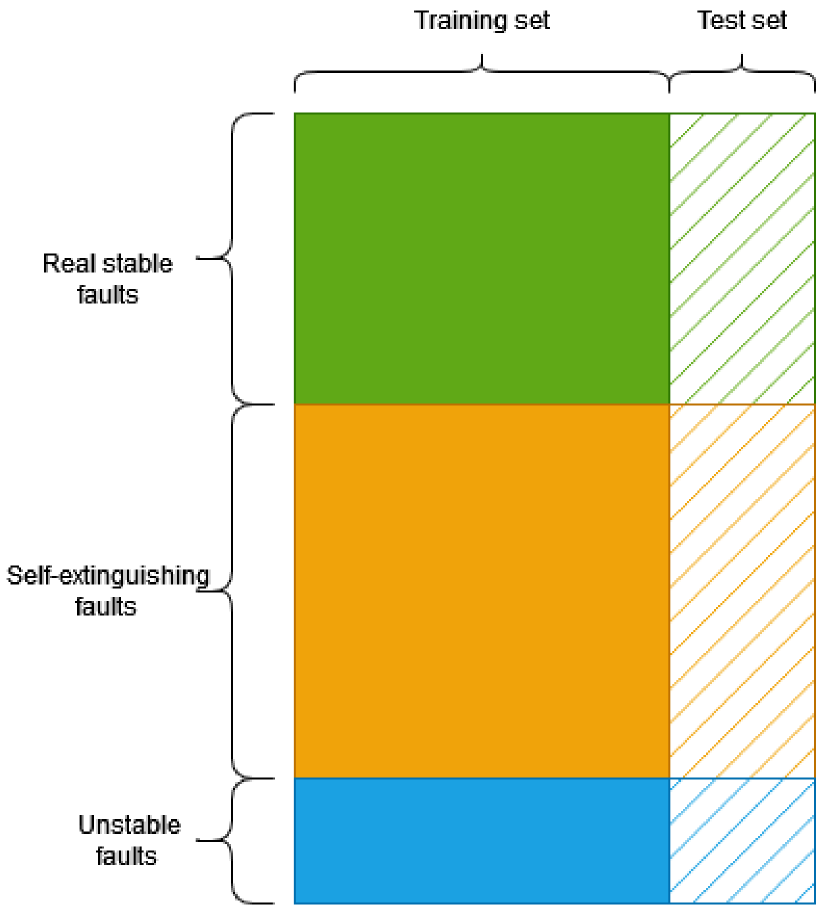


Figure 3.1: Description of the dataset separation needed for the suitable evaluation of the models

ing altered, so the ML model could see differently the same entry in each case, there is a clear division of the dataset in training, validation and test sets, in order to keep the data used for evaluation certainly unseen. Therefore, the stable faults that belong to the test sets needed for the two models in chapter 5 are also removed from the training and validation sets of the models developed in chapter 4.

Figure 3.1 depicts the separation through all over the whole real dataset. So, for the training and validation of the models concern the stable faults in chapter 4, only the solid green box is used. For the training of the stability rules model in chapter 5 the solid blue box is used, where as then for fitting the data to both that and the classification model the whole training set is used and for the evaluation of the two models the whole test set. The test set was randomly selected in advance. Additional stable faults that were initially labeled unstable by Alliander were added, as well as the 8 stable faults misclassified by that. This way it would be possible to compare the performance of the model developed there with the one currently in use by Alliander.

3.2. Machine Learning

ML constitutes a subset of artificial intelligence (AI). It solves problems based historical data and information, using algorithms and statistical models without the need of explicit programming and executing complex mathematical calculations in a fast way [37]. The main function of ML is to learn and recognize hidden patterns within the available data and then use these patterns for an output prediction or decision of a related event. The ML algorithms can be distinguished in two main categories, supervised and unsupervised learning and their difference is that for the former there exist labels in the training data subset [38]. Supervised learning can further categorized into regression and classification problems. The regression problem involves a continuous output, whereas classification problem involves a discrete output. Unsupervised learning mostly consists of clustering and association mining techniques. Clustering algorithms form various inherent subgroups based on the hidden relationships between the attributes of the observations, while association mining algorithms focus on finding suitable rules that express the relationships between variables accurately. In figure 3.2 a diagram with the ML subcategories is shown.

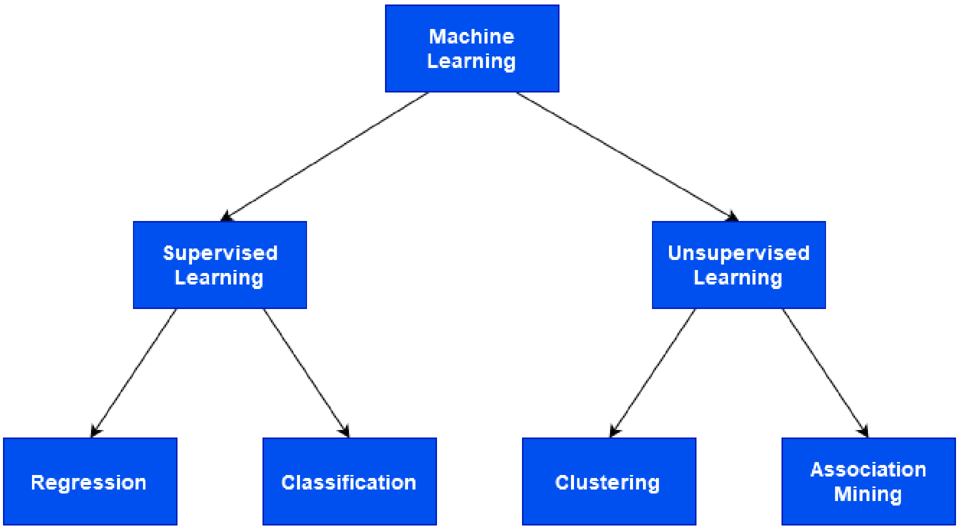


Figure 3.2: Machine Learning subcategories diagram

3.2.1. Supervised Machine Learning

A main problem that is intended to be solved is the classification of the fault type. The dataset used for this problem involves the three-phase current and voltage waveforms of the fault signals recorded from the SASensor, as well as their corresponding type of fault. This is actually the label of each data entry and the one that is going to be predicted by the model in the end. So, this constitutes a supervised learning problem.

In order to create such an appropriate classification model, first this needs to be fitted by a training set of data and then evaluated by a test set. So, the goal is to have a model with an accurate enough performance on the test set, which is the unseen set of data. These two sets must be split in advance and there must be no overlaps, so there is no leak in the fitting procedure.

For given N training samples $X = \{x_1, x_2, ..., x_N\}$, where $x_i \in \Re^m$ is the ith sample and each of these samples has m dimensions or features. Each training sample corresponds to a label $Y = \{y_1, y_2, ..., y_N\}$, where $y_i \in \{1, 2, ..., c\}$, with a certain class. Therefore, in the end the training data set is $\{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$. During the training procedure, such a machine learning model aims at discovering the relationship between the observations X and their labels Y. Subsequently, the performance of the fitted model is evaluated on the test set, the unseen data and in particular, the accuracy of the predicted results is checked compared to the actual labels. That way, the model's ability to generalize and performance on future real unseen data is measured.

Since the generalization of the model and the response in future data has high importance, there should be a suitable trade-off between bias and variance. A model that is fitted remarkably close to the training data, learning all the details and small irregularities of the dataset, including noise and various other random components, inevitably cannot be generalized properly. Thus, although it has an extremely high scoring performance on the training dataset, it has a poor scoring performance on the test dataset. This means that the model has high variance and this is called overfitting. In contrast, a model that learns only broad and loose relationships of the dataset, while missing the meaningful aspects, it has a high bias. This is called underfitting. [39]. Therefore, the goal is to have an harmonized trade-off between bias and variance of the model, so it can follow the useful and significant trends of the dataset, without entailing very trivial and noisy aspects of that.

3.2.2. Support Vector Machine

As seen from the literature review, SVM is a very suitable model for a classification model. It can use hyperplanes to separate the different classes optimally, not only for linearly but for non-linearly separable data as well. This is feasible due to the use of kernel functions, which can transform the feature space into a higher-dimensional space, such as cubic, quadratic etc where the data is linearly separable.

However, discovering a hyperplane that entirely separates the data is not the only criterion for a well-performed classification. In such a case, the sensitivity of the model is endangered, because even one single point can change the position of the hyperplane. Thus, the SVM finds such a hyperplane that does allow a few misclassifications during the fitting procedure by the training set, so it can perform better in the unseen data. That comes after an suitable determination of the bias/variance trade-off.

In Figure 3.3, a hyperplane is shown which can adequately separate the two datasets. As seen, the hyperplane here is a line, since the space is 2-dimensional. Correspondingly, the hyperplane is a plane in a 3-dimensional space and in general, for a m-dimensional space, the hyperplane is a subspace with $m-1$ dimensions. The decision boundary between the classes in the figure 3.3 corresponds to the line $w^T x_i + b = 0$, where w is the weight vector, b is the bias and x is the training sample data.

Here the data is non-separable, so there are some misclassifications [39]. In that case a new variable, the slack variable $\xi_i \geq 0$ is introduced, which is equal to the distance between the misclassified point x_i and the its respective margin hyperplane. When $0 \leq \xi_i \leq 1$, the sample is located between the hyperplane and the correct margin plane which means that there is no misclassification. However when $\xi_i > 1$, the sample is not classified correctly. The case of separable data is the special case when $\xi_i = 0$. So, the general equation for the margin hyperplanes is:

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \forall i = 1, 2, ..., N \tag{3.1}$$

where $y_i \in \{-1, +1\}$ here for the negative and positive class respectively.

The distance between the margin planes is called margin and it is equal to $d_1 + d_2 = \frac{2}{\|w\|}$, where d_1, d_2 correspond to the distance from the two planes to the hyperplane respectively. The objective function of the

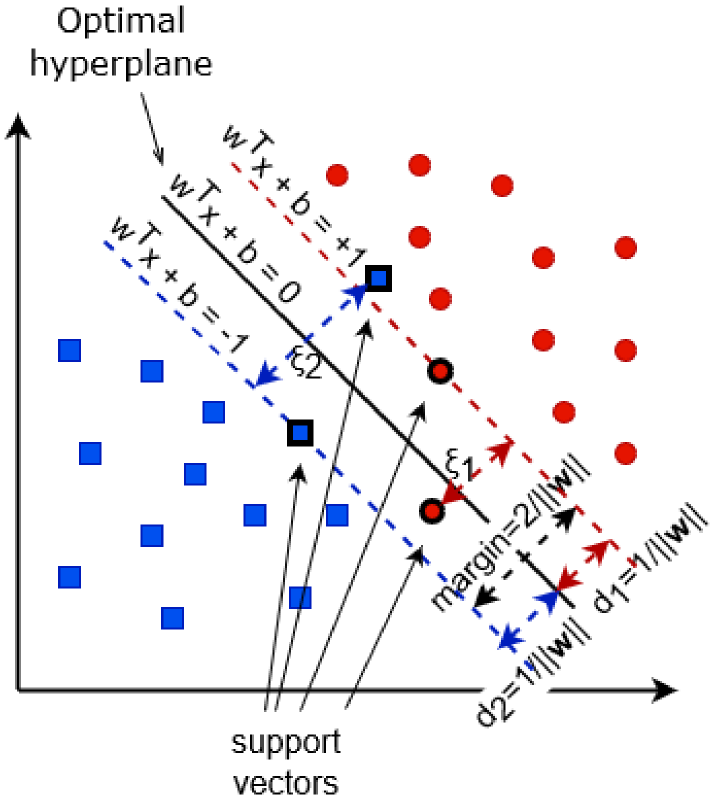


Figure 3.3: Diagram of the seperating hyperplanes using support vectors for a two-class classification problem

SVM equals:

$$\begin{aligned} &\min\ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi_i \\ \text{s.t.}\quad & y_i(w^Tx_i+b)-1+\xi_i \geq 0, \quad \forall i=1,2,...,N \\ & \xi_i \geq 0 \end{aligned} \tag{3.2}$$

where C is is the penalty factor for misclassifications.

As mentioned above, when the dataset samples are non-linearly separable ($\xi_i > 1$), SVM can use kernel functions so that maps the data in a higher-dimensional space, where the data may be linearly separable [39]. These kernel functions consist of the dot product of nonlinear functions ϕ and more precisely they are equal to $K(x_i,x_j)=\phi(x_i)^T\phi(x_j)$. As from Ranganathan [2] and Livani et al [19], the suitable kernel function in the case of fault classification is the radial basis function (RBF), which equals to:

$$K(x_i,x_j) = exp\left(-\frac{\|x_i-x_j\|^2}{\gamma}\right) \tag{3.3}$$

Therefore, using a SVM with a RBF kernel, in order to tune the model satisfyingly, it is crucial to determine the values of the parameters C from equation 3.2 and γ from 3.3. These parameters are called hyperparameters and the process of finding their values is called hyperparameter tuning. In particular, parameter C is the regularization parameter since it can control the trade-off between the classification margin and the classification error of the training dataset. Parameter C is proportional to the variance and inversely proportional to the bias of the classifier, so an extremely high value of C can lead to overfitting and a extremely low value can lead to underfitting. Parameter γ determines the degree of similarity between two training samples. A small γ results in high variance, low variance and an almost flat hyperplane, whereas a large γ results in low variance, high bias and a more curvy and spiky hyperplane. So, it has a huge influence on the separating hyperplane's position and shape. In conclusion, a simultaneous optimization of C and γ values must be done such that the model works appropriately.

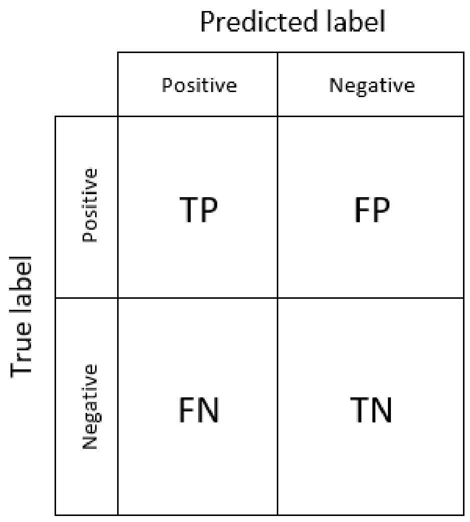
For the fault classification of these thesis there exist at least 3 different classes. Thus, a multi-class classification has to be applied which makes some aspects of SVM different. Since SVM was designed for binary classification, there exist two different techniques to convert it to multi-class classification [40]. The first is the "one-against-all" method where each class is trained against the rest of the classes. The second is the "one-against-one" method where each class is trained against each class separately. For the first approach the training of k classifiers and for the second approach the training of $k(k-1)/2$ classifiers are required, where k is the number of classes existed. So, the "one-against-all" is more computational efficient and interpretable, however the "one-against-one" is usually more accurate with kernel algorithms [41]. "One-against-all" is the most commonly used technique and it is the one preferred for this project.

3.3. Metrics for Evaluating the Model's Performance

3.3.1. Metrics for Evaluating the Classification Model's Performance

In this thesis project the f1-score is used for evaluating the performance of the classification model. F1-score derives from the combination of precision and recall which could also be used for specific evaluation of a classification model. In the following section the definitions of precision, recall and f1-score are introduced.

In figure 3.4 a confusion matrix for binary classification is presented, so these metrics could be explained better. Each sample can be classified as positive or negative. Each row of the matrix corresponds to a true class whereas each column corresponds to an predicted class. So, the number of positive instances correctly classified as positive is called True Positive (TP) while the number of negative instances correctly classified as negative is called True Negative (TN). Combined these two consist of the samples that are classified correctly. In contrast, the number of positive entries which have misclassified as negative is called False Negative (FN) and the number of negative entries which have misclassified as positive is called False Positive (FP). These two combined consist of the misclassifications of the model. With these definitions, the formulation of the metrics precision, recall and f1-score can be shown.



		Predicted label	
		Positive	Negative
True label	Positive	TP	FP
	Negative	FN	TN

Figure 3.4: Confusion matrix for a binary classification problem

Precision is defined as the ratio of the correctly classified positive samples to the total instances that are predicted to be positive [42]. It indicates the accuracy of the model in classifying the positive class and it is useful when FP is a higher concern than FN, such as in spam recognition. Its equation is given in equation 3.4.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{3.4}$$

Recall represents the ratio of the correctly classified positive samples divided by the total number of the samples that are actually positive [42]. It indicates the strength of a model to predict the positive class and it is useful when FN exceeds the importance of FP, such as in a disease diagnosis. Recall is also called sensitivity and because it is using only the fraction of entries that are positive is not that a standalone metric. Equation 3.6 shows the equation of recall.

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{3.5}$$

Precision and recall cannot represent separately the entire performance of a classifier because they focus mostly on the positive class covering each a half of the confusion matrix not in compliance. So, the f1-score metric is introduced which represents the weighted harmonic average of precision and recall, aiming at the optimal trade-off between these two metrics [43], [44]. That way f1-score is high when both precision and recall is high and low when both are low. It is designed so it works well with imbalanced data and its equation is provided in equation 3.6.

$$\text{F1-score} = 2\,\frac{\text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}} \tag{3.6}$$

When there is a multi-class classification problem, there is not an overall F1-score calculation. Instead, the evaluation becomes a one-vs-rest manner where each class is evaluated against all the other classes. So, each class success is rated in distinct steps, treating the rest of the classes as the negative one and subsequently turning the multi-class problem into a combination of binary classification problems [44]. For each class the precision, the recall and then the f1-score are calculated. For a better comparison a single value of f1-score is needed so the final result is an average of each class's score. There are three averaging techniques namely macro, weighted and micro. In particular, macro averaging is calculated by the unweighted mean of all the per-class f1 scores, weighted averaged f1 score is computed by the weighted mean of all the per-class respective scores and micro averaging is the global average where the sums of TPs, FNs and FPs are considered. For the evaluation of multi-class classification problems in this project the averaging method used is mostly the macro. It is suitable in situations like this with the given dataset that contains imbalanced classes, but regardless all the classes are treated equally and there is no distinction between highly and poorly populated classes [44]. The weighted averaging method is used as well for the final result, since there is a huge difference in the balance of the five different classes and in order to compare previous results where the same method was used.

3.3.2. Metrics for Evaluating the Model's Performance on Calculating the Loop-Impedance

In the scope of this project is after classifying the type of the faults to also provide an accurate fault loop-impedance so the fault location could be found. Thus, there is need to evaluate how close is the calculated by the thesis model loop-impedance compared to the one given by Alliander. The main metrics used for this evaluation are the root mean square error (RMSE), the mean absolute error (MAE) and the 90th percentile of the absolute error.

$$RMSE = \sqrt{\frac{1}{N}\sum_{n=1}^{N}(y_i - \hat{y}_i)^2} \tag{3.7}$$

$$MAE = \frac{1}{N}\sum_{n=1}^{N}|y_i - \hat{y}_i| \tag{3.8}$$

Their equations of the first two are presented in equations 3.7 and 3.8 respectively, where y_i is the impedance value as calculated by Alliander with the Vision FaultFinder, $\hat{y}_i$ is the predicted impedance value by the model developed in this thesis and N is number of total data in the specific set. Ideally the error should be zero, so basically the goal is to find sufficiently low error on the set of faults that are evaluated each time. The difference between the two first metrics is that RMSE is penalizing more the further points or outliers than MAE, so it takes into consideration even the small number of data that lead to very bad results. The use also of the 90th percentile of the absolute error helps into detecting where the huge part of the errors stands (lower 90%) and that indeed the rest of it could be outliers.

3.4. Fault Segmentation Algorithm

Each signal entry can be divided into three parts, the pre-fault state, the fault state and the post-fault state. The part with the important information about the fault and its type is the fault state. So as Ranganathan also showed in [2], it would be more beneficial to isolate and use only the fault state for further analysis. Since the ML method used is SVM, it is more suitable for the performance of the model to use features that reflect closer to the faulted part, that consists of the information regarding the type of fault.

Segmentation is gaining also great significance from the fact that in the duration of the fault, more than one types could exist due to the nature of faults in reality to evolve into a more serious type. That is the main reason why the segmentation algorithms are not stressed in the literature where the simulated faults used as data do not possess evolving stages. Usually only the late pre-fault and early fault stages are used for draining information about the fault type which in case of an evolving fault would lead to a not correct result. It is more useful to use the most serious fault for the fault location, since it is more accurate the calculation of the estimated fault location in the case of a multi-phase fault. Thus, applying a suitable segmentation algorithm ensures that the most serious fault stage is isolated and used further for the fault location estimation.

Figure 3.5 shows a three-phase fault. This is the final type of fault, since it begins as a single-phase fault and then progresses into a two-phase fault before it evolves into a three-phase fault. In that case the fault location could calculated more accurately from the three-phase fault part. In the rest of this section the theory behind the algorithm used is explained and the different implementations used are presented.

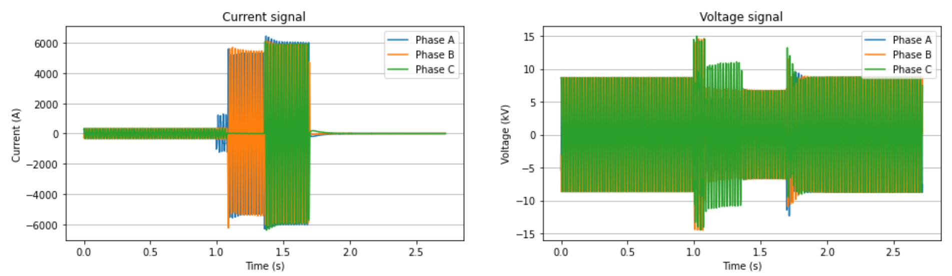


Figure 3.5: Fault that starts as a single-phase fault, then evolves to a two-phase fault and in the end a three-phase fault

3.4.1. Binary Segmentation Algorithm

For segmenting the fault signal and ending up to the final fault stage the binary segmentation algorithm is used. This is an offline change point detection method which helps detect change points such as major disruptions in a time series and more specifically the fault inception and the fault clearing time for a fault signal [45], [46]. For a multivariate non-stationary time series $y_1, y_2, ..., y_T$ that consists of T samples, if the number of change points is K, then there are possible change positions $t_1 < t_2 < ... < t_K$ which partition the signal into $K-1$ sections ($K+1$ sections if the start and the end of the signal entry are included). The i-th segmented section consists of $(y_{t_i}, y_{t_{i+1}})$.

The objective function of this optimization problem can be represented by the equation 3.9 as seen below.

$$\begin{aligned}\hat{t}_1, \hat{t}_2, ..., \hat{t}_K &= \arg \min_{t_1,...,t_K} V(t_1, t_2, ..., t_K) \\ \text{s.t.} \quad V(t_1, t_2, ..., t_K) &:= \sum_{k=0}^{K} C\left(y_{t_k...t_{k+1}}\right)\end{aligned} \tag{3.9}$$

where $C(y_{t_k...t_{k+1}})$ is the cost function. By convention the zero changepoint of the time series is $t_0 = 0$ and the last changepoint is $t_{K+1} = T$. The cost function used could be among others the least absolute deviation, least square deviation, gaussian process change, kernelized mean change [45]. Here the least square deviation is selected and is given in equation 3.10.

$$C\left(y_{t_k...t_{k+1}}\right) = \sum_{t=t_k}^{t_{k+1}-1} \left\| y_t - \bar{y}_{t_k...t_{k+1}} \right\|^2 \tag{3.10}$$

where $\bar{y}_{t_k...t_{k+1}}$ is the average value of the part $(y_{t_k}, y_{t_k+1}, ..., y_{t_{k+1}-1})$ of the signal. Eventually the binary segmentation aims to find the required changepoints that optimally split the whole signal based on the best mean-shifts.

3.4.2. Segmentation Algorithm Implementations

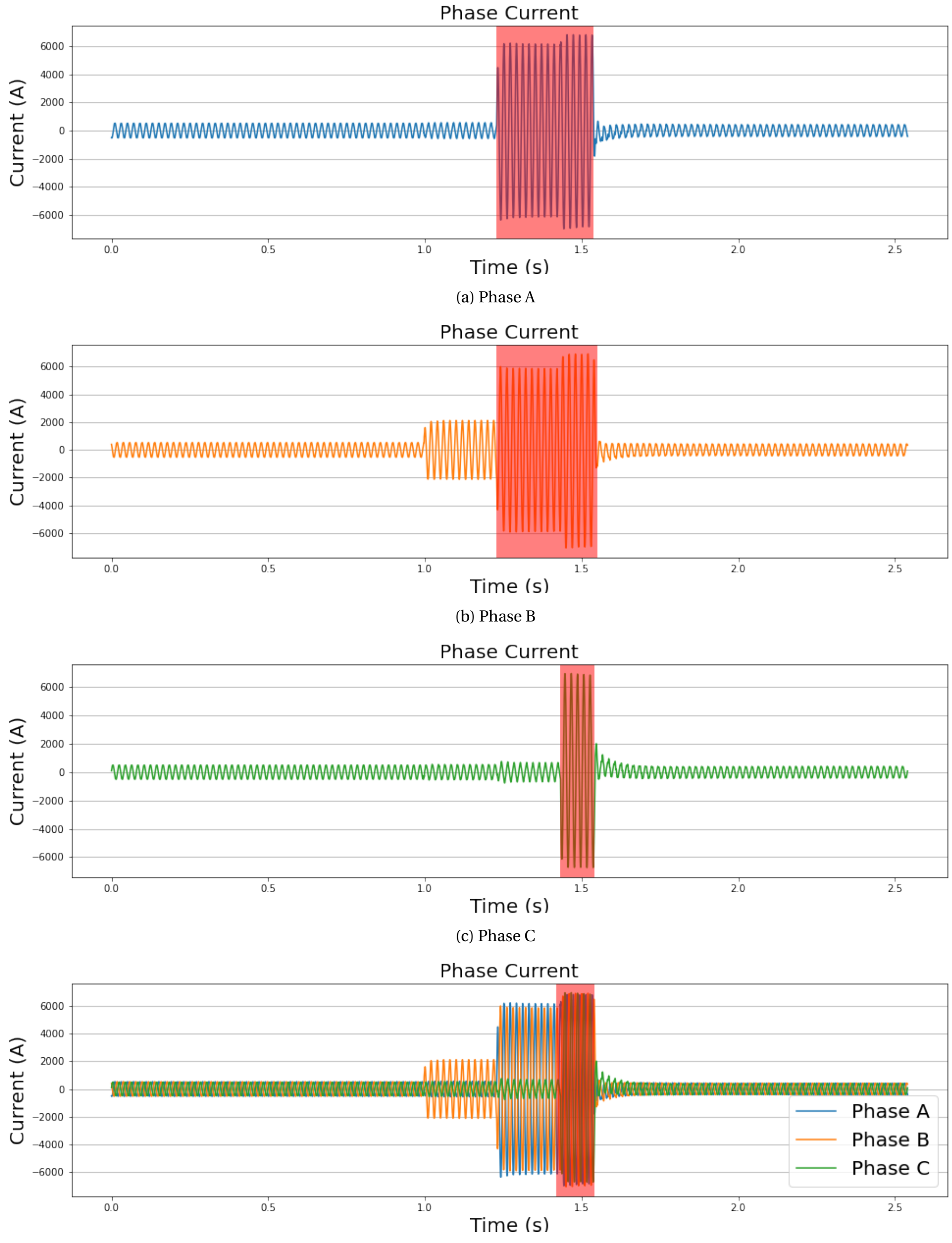
The different ways of implementing the binary segmentation algorithm are presented in this section. The aim is to isolate the final serious part of the fault before its clearing. For the two model developments in chapters 4 and 5, two different approaches are applied. For the model in chapter 4 a combination of univariate segmentation and peak techniques with two breakpoints are used whereas in chapter 5 a multivariate segmentation with three breakpoints is used. For both the implementations the same objective function is applied, since it is suitable for both univariate and multivariate input data. In the following subsections each of these techniques is further explained.

Univariate segmentation algorithm

For the first case, the binary segmentation algorithm is applied in the current signals of each phase providing three sets of two breakpoints which correspond to the most eminent change in mean value of the signal, similar to what Ranganathan used in [2]. The absolute value of each signal is used so the zero mean of the sinusoidal signal is eliminated. In order to make sure that the segmentation is going to be correct, the largest value of the first breakpoints is selected and the average of the second breakpoints.

Figure 3.6 shows an example of this univariate segmentation technique. The calculated breakpoints for each phase are shown, as well as the final result. In general, since Ranganathan in [2] showed that a similar segmentation technique on similar dataset and model was sufficient in isolating the final section of the fault, it is quite promising for the performance of the model in this project.

Regarding the self-extinguishing faults, this algorithm cannot always detect suitable changepoints, since they are either too short or the consists of a lot of similar disturbances in the signal. So, when the difference between two break points is less than two periods, the whole signal is selected instead of a section. The features used for the model later would be selected appropriately so the a long segmented part indicates a self-extinguishing fault. For instance, using the average value of the current at the whole signal of a self-extinguishing fault, the influence of the sudden and short-time current increase would be deteriorated.



(a) Phase A

(b) Phase B

(c) Phase C

(d) Final three phases

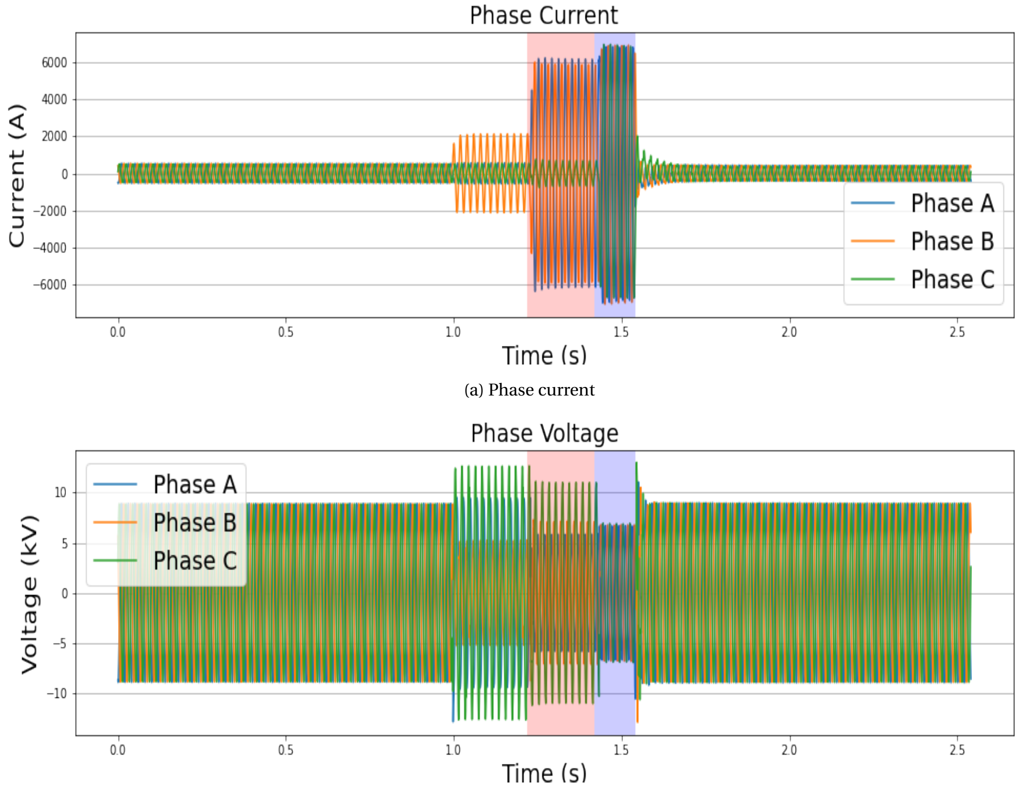
Figure 3.6: Univariate segmentation

Multivariate segmentation technique

For the second case, the binary segmentation algorithm is implemented in all the current and voltage signals of a data entry, as a multivariate technique. Once again the absolute value of the signals is used for the elimination of the zero mean. The algorithm returns three breakpoints instead of two, that correspond to the most notable mean-shifts in the signal and can create two segmented sections, from the first to the second changepoint and from the second to the third changepoint.

Figure 3.7 displays an example of the multivariate segmentation. The two resulted segmentations are important for the model in chapter 5, since signal-wise may the last fault section be selected by the previous algorithm. However if this is an unstable fault and if there is before another different fault section, that should also be considered for the stability and the fault location. More details how the selection between the two sections is provided in chapter 5.

Again, all the segmented sections that last less than two periods are discarded. Even one section valid is sufficient for the model, but when both are discarded then the whole signal is selected as before. This is mostly applied to self-extinguishing faults to promote them by the smoothness that the entire signal would bring to the model compared to the segmented relatively small sections.



(a) Phase current

(b) Phase voltage

Figure 3.7: Multivariate segmentation

3.5. Summary

In this chapter a required theoretical background knowledge was given essential for the development of the models in the following chapters. The nature of the dataset used was presented and the way is split so the models could be trained and evaluated fairly. The necessary theory of ML was shown with a deeper description regarding the SVM as it is the preferred ML method and the main evaluation metrics for the classification and the loop-impedance calculation models were depicted. In the end, the univariate and multivariate segmentation techniques were defined which both are based on the binary segmentation algorithm.

4

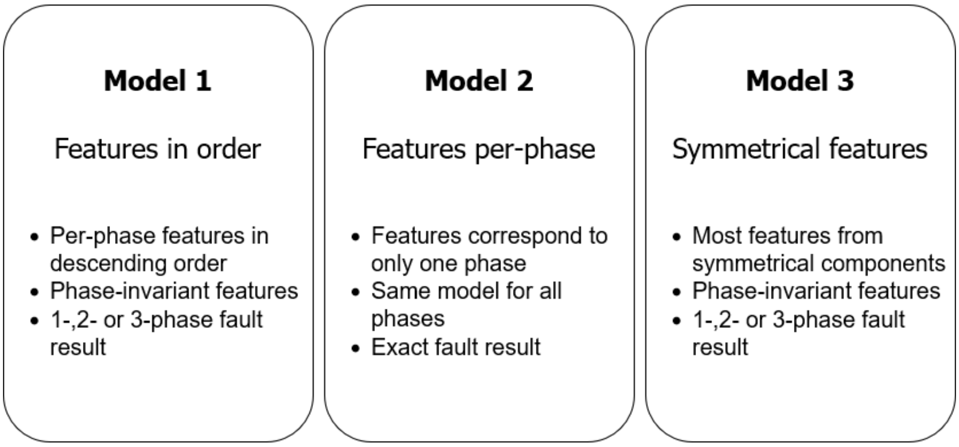
Classification Models with Explainable Features for Stable Faults

For the development of a machine learning (ML) model, it is important to have a solid feature set that really captures the essence of the problem, which usually involves various characteristics of the existing data set. So, this is the way to translate the information derived from the data into a meaningful input set for the model. In this project, the aim is to develop explainable classification models that have explainable features. That makes the task for the development of the feature set even more important and constrained, since it is the main source of explainability.

4.1. Development of Explainable Features for Stable Faults

The features reflect on specific property of the data that compares each entry among the others. So, there is a need for features that can identify the differences among the various classes that would distinguish them more effectively. In the following part is explained in more detail the process of developing three models with different feature sets.

4.1.1. Development of Three Models



Model 1

Features in order

- Per-phase features in descending order
- Phase-invariant features
- 1-,2- or 3-phase fault result

Model 2

Features per-phase

- Features correspond to only one phase
- Same model for all phases
- Exact fault result

Model 3

Symmetrical features

- Most features from symmetrical components
- Phase-invariant features
- 1-,2- or 3-phase fault result

Figure 4.1: Diagram of the three models

The decision was to develop three different models, so there will be a comparison of the results of each model. The ML model is similar for all three versions, since for all of them the Support Vector Machine (SVM) algorithm for classification is used. Three different approaches were depicted about the feature set development, therefore the main difference among them derives from the way feature set is shaped for each model. The majority of models in literature are using explicit indication in which exact phase the fault occurs, such as [18], [32], [33], [20], [19], whereas fewer examples such as [2] a model with no exact faulted phase is being used. In such a case, incorporating symmetry is substantial since the same faults with a phase rotation should lead to the same result. For example, a single-phase fault with the fault in phase A, should provide the same information and be recognized as exactly the same single-phase fault, even if with the phase rotation the fault occurs in phase B or phase C. Thus inserting symmetry to the model it can lead to a more generalized and a less prone to overfitting model as Bergman depicts in [47]. This can be done by developing phase-invariant features. Since Alliander's current classification model uses a non explicit phase fault model, the Digital Fault Recordings (DFRs) and the excel files with the DFR entries do not posses that information. So, any per-phase classification model would need an additional preproccessing stage, where the data would be further categorized in phase-dependent labels with a manual inspection.

The first distinguishing step among the three models developed is that the one is a per phase model and the other two are incorporating all the three phases at the same time. The former model was developed based on what the majority in literature uses, despite the need for a prepossessing level. Each of the three phases trains the same ML model separately, so that makes the model work appropriately under phase permutation. The latter models are not per-phase models but they are both symmetrical. The difference between them is depicted on the way the features used are getting phase invariant. In the one the phase decoupling is done using the features of each phase in an order (e.g. maximum to minimum value) replacing the features used in [2] with corresponding more explainable ones. The symmetry of this model with the features in order is forced by the metrics used. In particular, using a rank of the values three different features are created corresponding to the maximum, middle and minimum value. Thus, especially the centre value loses its meaning because there is not a specific definition that determines the distance between this and the minimum or maximum value. Also, this is always dependent on those as its value is limited by them both in the upper and lower boundary. Another property of that model is the discontinuity of the values and the considerable jumps among the values of these three levels. In the other model, mainly the symmetrical components are used for the feature extraction, which injects a more pure symmetry into the model due to their nature and consist of continuous values. The other features are developed that way so to be standalone symmetrical features. Figure 4.1 shows some of the key characteristics of each model.

It is important to mention some assumptions made and followed by all the models' feature extraction. First, all the features are developed after the segmentation algorithm, so they are related to the segmented part of the signal entry which essentially corresponds to the suspected fault period. Second, all the features, except for the average of active power for the symmetrical model and the peak to peak current value in the per-phase model, are developed based on the sliding window Fourier transform (FT). Lastly, the use of positive and negative sequences as symmetrical is valid, due to the fact that the voltage and current entry values in the DFR files follow the sequence of A->B->C. That makes sure that the model is symmetrical under the phase permutations A->B->C->A of the data.

4.1.2. Features based on the Short-time Fourier Transform Results

As mentioned above, the majority of the features is developed based on the Short-time FT (STFT). Using this, the required frequencies can be isolated per period and used for a more general behaviour of the signal. This is in a way similar to the WT, but with quite less adaptability since a single window size is used. The focus on the explainability brings STFT in more prominent position for use in this specific project, though. Furthermore, the STFT for the fundamental frequency provide similar information with the root mean square (RMS) value of the same period, as a single period waveform is expressed with a single magnitude. However using the former some additional benefits are offered. Since the result of FT provides absolute value and angle for each period, it is quite more informative than the per-period RMS value and can help grasp a better image of the fault signal. More importantly, it can be used to define the per-period impedance, reactance, resistance and symmetrical components, which cannot be calculated through the real signal.

Mostly, the sliding window FT is being used for capturing the fundamental frequency component for the voltage and current signals and then shaping accordingly the impedance and the symmetrical components.

For all the features, the sliding window consists of 80 samples, which corresponds to one period. However the step is not the same for all the cases. Mainly a step of 80 samples is used, thus there is no overlap of windows and that means a per-period analysis. Only for the features concerning the third harmonic a step of 1 sample is used. Such features are essential for the transient parts and there, a better resolution seems like a more reasonable approach. This fundamental component is a complex component that consists of the magnitude and angle of the signal for a specific period. This holds when the signal is pure sinusoidal and that's why the faults without a sufficient sine wave (unstable) cannot lead to reliable results regarding the fault loop impedance. In the chapter 5, new rules to distinguish better stable faults from unstable ones regarding their waveform are developed. The per-period complex voltage fundamental component divided by the corresponding current component, can lead to the per-period impedance calculations. The value of that complex number resulted by this ratio constitutes directly the impedance value, the imaginary part corresponds to the reactance and the real part to the resistance, as shown in equations 4.1, 4.2 and 4.3 correspondingly. In those equations, V and ϕ_V correspond to the fundamental voltage magnitude and angle for a period, whereas I and ϕ_I correspond to the fundamental current magnitude and angle for a period. In equation 4.1 the absolute value of the per-period impedance is calculated, so only real values are used.

$$|Z| = \left| \frac{V\angle\phi_V}{I\angle\phi_I} \right| \tag{4.1}$$

$$X = Im\left(\frac{V\angle\phi_V}{I\angle\phi_I}\right) \tag{4.2}$$

$$R = Re\left(\frac{V\angle\phi_V}{I\angle\phi_I}\right) \tag{4.3}$$

In a similar way, from the per-period complex components the symmetrical components can be formed for the voltage and the current signals, as seen in the equation 4.4 for the current symmetrical components, where I_0, I_1, I_2 are the zero, positive and negative current components respectively, $I_a\angle\phi_{I_a}$, $I_b\angle\phi_{I_b}$, $I_c\angle\phi_{I_c}$ correspond to the pair fundamental current magnitude and angle for phase a, b, c, correspondingly and $\alpha = 1\angle 120°$.

$$\begin{pmatrix} I_0 \\ I_1 \\ I_2 \end{pmatrix} = \begin{pmatrix} 1 & 1 & 1 \\ 1 & \alpha^2 & \alpha \\ 1 & \alpha & \alpha^2 \end{pmatrix} * \begin{pmatrix} I_a\angle\phi_{I_a} \\ I_b\angle\phi_{I_b} \\ I_c\angle\phi_{I_c} \end{pmatrix} \tag{4.4}$$

In addition, STFT is also being used for the third harmonic frequency component, so the features for the transient phenomena to be extracted. This way, changes of an event's state are captured, such as the fault inception and fault clearing. This effectively consists of the replacement of the Wavelet transform (WT) coefficient with more explainable way. The third harmonic is a harmonic that does not pass from the low voltage (LV) to the medium voltage (MV) through the LV/MV transformer. So, the existence of a third harmonic component in current or voltage is a good indication of a disturbance inside the MV network and the higher is that value the higher is the probability for a fault in that phase. Figure 4.2 shows the current and voltage signal of a single-phase fault as well as the third harmonic of the current and the voltage. Figure 4.3 depicts the corresponding plots for a three-phase fault.

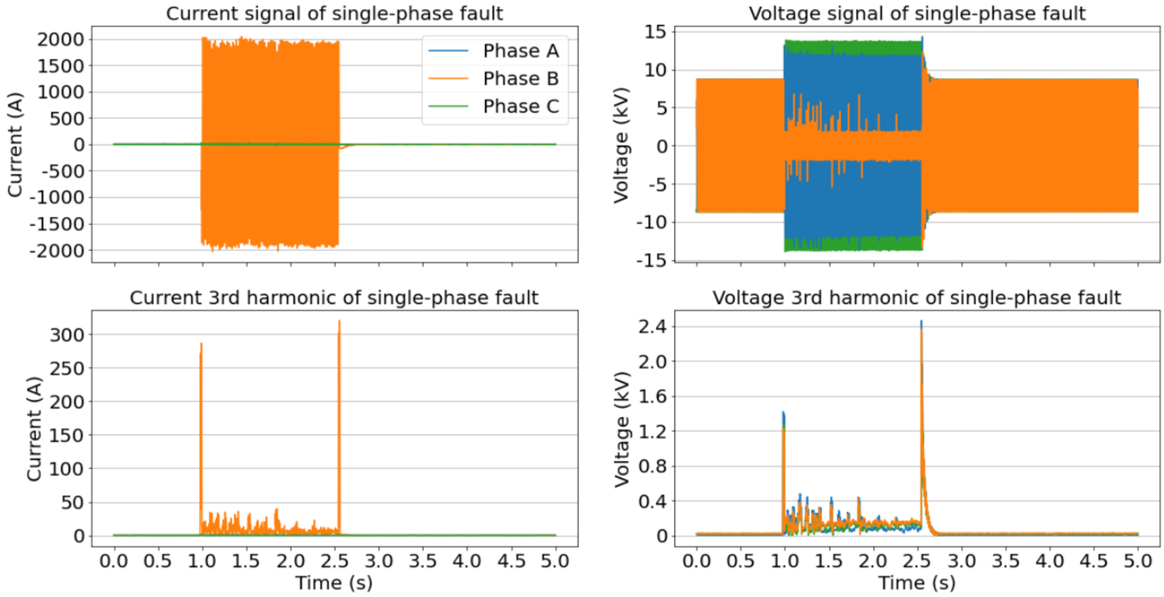


Figure 4.2: Current, voltage signal third harmonic of the current and the voltage of a single-phase fault

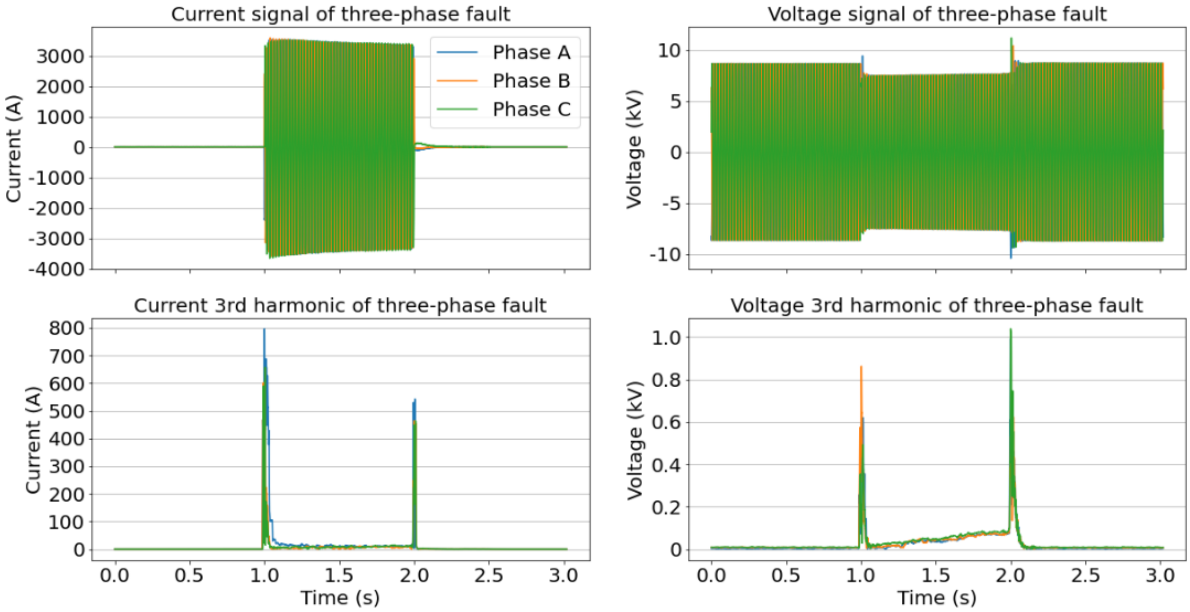


Figure 4.3: Current, voltage signal third harmonic of the current and the voltage of a three-phase fault

4.1.3. Features based on the Symmetrical Components Ratios

In the following sections, some features will be presented that derived from symmetrical components' ratios. As shown before, based on the STFT the per-period symmetrical components could be shaped for the current and the voltage. Taking one step forward, for the feature selection of this thesis' model, features coming from different combinations of symmetrical components ratios are used. This is done based on Gu and Provoost [3] where the ratio I_2/I_1 is used as a criterion for the distinction between two- and three-phase faults. Here, also the ratios $I_0/I1$, V_2/V_1 and V_0/V_1 are used as a basis for the further development of features. The thought behind this is that these ratios first have an expected value which is per unit and normally would be between 0 and maximum 2. Second, for each type of fault there is a specific area of the values anticipated for each one of these per-period ratios. In particular, these approximate per-period values are shown in table 4.1. Therefore, the features developed in the following sections based on these per-period ratios would have an expected value that would reflect to the correct type of fault. That makes them more explainable and interpretable if used and this is a very essential advantage for this project given the importance of explainability.

Table 4.1: Anticipated values of the symmetrical components ratios for each type of fault

Fault type	I_0/I_1	I_2/I_1	V_0/V_1	V_2/V_1
1	~1	~1	~1	~0
2	~0	~1	~0.5	~0.5
3	~0	~0	~0	~0

4.1.4. Statistical Metrics used for the Feature Extraction

Before presenting in detail the features extracted, it is important to mention the metrics that are used for that. That aspects that determine the decision over the specific metrics used are the need for explainability, the use of segmentation and of course the domain of the data. Since explainability comes from the feature engineering is significant to used not complicated metrics. The segmentation focuses on the fault duration, so the metrics used do too and finally, the domain of the data, for example if it is a sine or sliding FT, influences the metrics used in the end.

The statistical metrics used are the mean or average value, the minimum value, the peak to peak value and the energy of the signal. In particular, energy is not the same as the active power of the signal and it consists of the summation of all the points of the signal squared, as shown in the equation 4.5.

$$energy = \sum_{i=1}^{N} x(i)^2 \tag{4.5}$$

4.1.5. Features from the Voltage signal

The extraction of features from the voltage was only done by using the sliding window FT. Mainly, the average of the segmented part was the feature used with some variations based on the model. For the sorted features model the mean value of the phase voltages of the segmented signal descending order *V_wft_mean* is used, with 1 named the maximum value, 2 the middle value and 3 the minimum. Also, the energy of the transient FT for the third harmonic *V_wft_harmonic_energy* of the voltage signal again in descending order. The mean value of the symmetrical components' voltages *V_0_mean*, *V_1_mean*, *V_2_mean* is used in both sorted and symmetrical features model. For the per-phase model, the mean *V_wft_mean* and the minimum value *V_wft_min* of the per-phase segmented voltage is used. Also, the energy of the transient FT for the third harmonic *V_wft_harmonic_energy* through the segmented voltage signal duration. In addition for the symmetrical model, two ratios of the symmetrical components segmented voltage mean values are used. *V_2_mean/V_1_mean* corresponds to the division of the average values of the negative to the positive voltage components and *V_0_mean/V_1_mean* corresponds to the division of the average values of the zero to the positive voltage components. Table 4.2 sums up the features derived from the voltage signals.

Table 4.2: Description of features derived from the voltage signal

Feature	Description
V_wft_mean_1 V_wft_mean_2 V_wft_mean_3	Maximum, middle and minimum order for the mean value of the windowed FT of the phase voltage
V_wft_harmonic_energy_1 V_wft_harmonic_energy_2 V_wft_harmonic_energy_3	Maximum, middle and minimum order for the energy of the third harmonic windowed FT component of the phase voltage
V_wft_harmonic_energy	Energy of the third harmonic windowed FT component of the phase voltage
V_0_mean V_1_mean V_2_mean	Mean value of the windowed FT of the zero, positive, negative voltage sequence
V_wft_mean	Mean value of the windowed FT of the phase voltage
V_wft_min	Minimum value of the windowed FT of the phase voltage
V_2_mean/V_1_mean	Ratio of negative sequence voltage to positive sequence voltage mean values
V_0_mean/V_1_mean	Ratio of zero sequence voltage to positive sequence voltage mean values

4.1.6. Features from the Current signal

In contrast with the features extracted by the voltage signals, the extraction of features from the current was not only done by using the sliding window FT, but also the original signal. First, in a similar way to the feature extraction from voltage, the mean value of the phase currents *I_wft_mean* of the segmented signal and the energy of the transient FT for the third harmonic *I_wft_harmonic_energy* of the segmented current signal both in descending order are used for the sorted features model. Also, in both these models, the mean value of the symmetrical components' currents *I_0_mean, I_1_mean, I_2_mean* is used again. For the per-phase model, similar to the voltage, the mean *I_wft_mean* and the minimum value *I_wft_min* of the per-phase segmented current and the energy of the transient FT for the third harmonic *I_wft_harmonic_energy* of the segmented current signal duration are used. For the symmetrical model, again two ratios of the symmetrical components segmented current mean values *I_2_mean/I_1_mean* and *I_0_mean/I_1_mean* are used, as the corresponding voltage ones.

However, there are two additional features that do not correspond to any of the voltage features. Firstly, based on the current sinusoidal signal, the peak-peak value *I_peak_peak* of the segmented duration is used in the per-phase model. Secondly, from the mean values of the segmented current signal the *I_wft_comparison* feature is developed as shown in the equation 4.6 and used in both sorted and symmetrical models. This has been derived based on a similar current feature in [2] where the root mean square values were used, instead of the sliding FT magnitude of the fundamental component.

$$I\_wft\_comparison = 2 * \frac{I\_wft\_mean\_1}{I\_wft\_mean\_2} - \frac{I\_wft\_mean\_2}{I\_wft\_mean\_3} - \frac{I\_wft\_mean\_3}{I\_wft\_mean\_1} \tag{4.6}$$

This feature is expected to have a value $I\_wft\_comparison >> 0$ when there is a single-phase fault, since the *I_wft_mean_1* is predominantly the higher component among the others. Concurrently, if there is a two-phase fault we expect both *I_wft_mean_1* and *I_wft_mean_2* to have high values and the *I_wft_mean_3* quite lower one, so the value of the feature should be $I\_wft\_comparison << 0$. Lastly, a value $I\_wft\_comparison \approx 0$ corresponds to a three-phase fault (or no fault) since all the magnitudes should have approximately the same values. Table 4.3 sums up the features derived from the current signals.

Table 4.3: Description of features derived from the current signal

Feature	Description
I_wft_mean_1 I_wft_mean_2 I_wft_mean_3	Maximum, middle and minimum order of the mean value of the windowed FT of the phase current
I_wft_harmonic_energy_1 I_wft_harmonic_energy_2 I_wft_harmonic_energy_3	Maximum, middle and minimum order for the energy of the third harmonic windowed FT component of the phase current
I_wft_harmonic_energy	Energy of the third harmonic windowed FT component of the phase current
I_0_mean I_1_mean I_2_mean	Mean value of the windowed FT of the zero, positive, negative current sequence
I_wft_mean	Mean value of the windowed FT of the phase current
I_wft_min	Minimum value of the windowed FT of the phase current
I_2_mean/I_1_mean	Ratio of negative sequence current to positive sequence current mean values
I_0_mean/I_1_mean	Ratio of zero sequence current to positive sequence current mean values
I_peak_peak	Peak to peak value of the phase current signal
I_wft_comparison	Metric based on the ratios between the mean values of the windowed FT for the phase currents

4.1.7. Features from the combination of Voltage and Current signals

For the sorted and per-phase features models, impedance-, reactance- and resistance-based features are developed. More precisely, for the first model, the average of each components' segmented part *Z_wft_mean*, *X_wft_mean*, *R_wft_mean* and then sorted in a descending order. For the second model, also the average and the minimum value *Z_wft_min*, *X_wft_min*, *R_wft_min* are calculated and used, whereas in symmetrical model the mean value of the active power *P_ft_mean* of the fault's segmented part is used.

In order to form these features, the per-period FT of the voltage and current signals are used, as mentioned above, so the impedance, reactance and resistance are calculated as in the equations 4.1, 4.2 and 4.3 correspondingly. The active power is simply calculated by the per-period fundamental magnitudes of all the phases' voltages and currents, as in the equation 4.7.

$$P\_ft = V_a * I_a + V_b * I_b + V_c * I_c \tag{4.7}$$

Table 4.4 sums up the features derived from the combination of voltage and current signals.

Table 4.4: Description of features derived from the combination of voltage and current signal

Feature	Description
Z_mean_1 Z_mean_2 Z_mean_3	Maximum, middle and minimum order of the mean value for the impedance
X_mean_1 X_mean_2 X_mean_3	Maximum, middle and minimum order of the mean value for the reactance
R_mean_1 R_mean_2 R_mean_3	Maximum, middle and minimum order of the mean value for the resistance
Z_mean X_mean R_mean	Mean value of the windowed FT for the impedance, reactance and resistance
Z_min X_min R_min	Minimum value of the windowed FT for the impedance, reactance and resistance
P_ft_mean	Mean value of the active power calculated based on windowed FT of phase currents and voltages

4.2. Model Development

In the previous section, the development of the three different models from feature set perspective was presented. Table 4.5 sums up the features developed for each model separately. There is still room for the final development for each of these models. First, the final feature set that represents better the dataset and the classification models has to be selected for each one of them. This is called feature selection. Filter methods such as analysis of variance (ANOVA) and mutual information are going to be used in order to evaluate the feature selection process. Second, there must be a hyper-parameter tuning. As mentioned in the chapter before, since the data need to be separated non-linearly, a radial basis function kernel is used to transfer the data points into higher dimensions where they could be separated. As a result, there are two hyper-parameters, namely C and γ, which their values need to be determined, so each model can operate properly. Finally, cross-validation is used to validate the performance of the models, after the fitting procedure and before tested on the very last unseen data. This is a good way, to evaluate the generalization of a model whilst in the fitting stage, so it can be adjusted accordingly before being evaluated by the test set.

Feature selection, hyper-parameter tuning and cross-validation constitute all parts of high importance for the model development. In order to acquire in the end a sufficiently generalized model though, it is better if these procedures are performed simultaneously and hence, a pipeline framework is created to help with that with a search all over the parameter's values. This way each of these steps do not target a locally optimal solution that would add more bias in the model but rather a more general solution, since each step would influence the next one to perform well. Otherwise, if all these procedures would be executed separately in sequence, the following procedures would depend on the finalised results of the previous ones that possibly would not lead their optimal solution. In the following parts of this section, feature selection, hyperparameter tuning and cross-validation are further explained.

Table 4.5: Summary of features developed for each model

Model 1	Model 2	Model 3
V_wft_mean_1	V_wft_min	V_0_mean
V_wft_mean_2	V_wft_mean	V_1_mean
V_wft_mean_3	I_wft_max	V_2_mean
I_wft_mean_1	I_wft_mean	I_0_mean
I_wft_mean_2	V_wft_harmonic_energy	I_1_mean
I_wft_mean_3	I_wft_harmonic_energy	I_2_mean
V_0_mean	I_peak_peak	V_2_mean/V_1_mean
V_1_mean	Z_mean	V_0_mean/V_1_mean
V_2_mean	Z_min	I_2_mean/I_1_mean
I_0_mean	X_mean	I_0_mean/I_1_mean
I_1_mean	X_min	I_wft_comparison
I_2_mean	R_mean	P_ft_mean
V_wft_harmonic_energy_1	R_min	
V_wft_harmonic_energy_2		
V_wft_harmonic_energy_3		
I_wft_harmonic_energy_1		
I_wft_harmonic_energy_2		
I_wft_harmonic_energy_3		
I_2_mean/I_1_mean		
I_wft_comparison		
Z_mean_1		
Z_mean_2		
Z_mean_3		
X_mean_1		
X_mean_2		
X_mean_3		
R_mean_1		
R_mean_2		
R_mean_3		

4.2.1. Feature Selection

Feature selection is the procedure of reducing the number of the features when building a predictive model, keeping the most important ones, hence improving the generalisability and reducing the variance of a model. According to Pal et al [48], the SVM classification accuracy is affected by the number of input variables. In particular, the accuracy of such a model is substantially reduced by using additional features and hence higher dimensions. This problem is called “curse of dimensionality” and also can lead to overfitting, so an underwhelming generalization of the model [49]. Therefore, since the performance of a SVM classification model depends on the dimensionality and the size of the dataset, the feature selection’s importance is growing. Added to the accuracy of the classification, feature selection leads to a lower computational cost and implicitly to a more interpretable model.

As depicted in figure 4.4, there are various feature selection techniques [50]. First they can be categorized into supervised and unsupervised methods, based on the use or not of target variables. Unsupervised feature

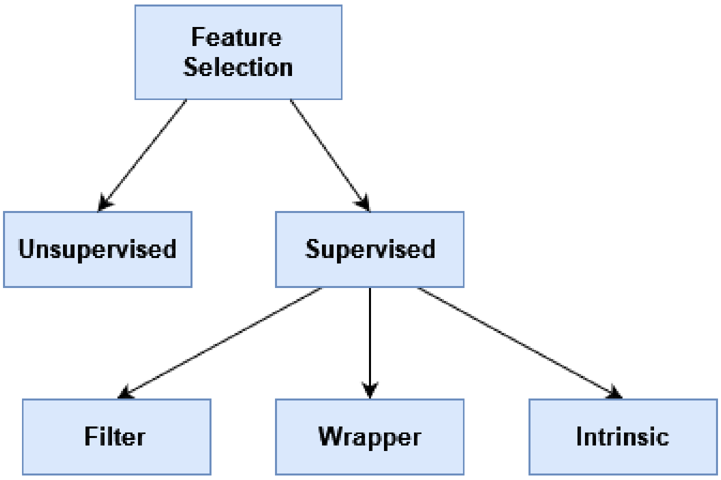


Figure 4.4: Feature selection techniques

selection techniques use correlation in order to remove redundant attributes with unlabeled data, whereas supervised techniques remove the irrelevant variables taking into account the labeled data. The supervised feature selection techniques are further distinguished into filter, wrapper and intrinsic methods [50], [49]. The filter methods use statistical techniques such as ANOVA and mutual information, to evaluate the relationship between each feature and the target and then based on that they rank the importance of the features [51]. The wrapper methods develop different subsets of input feature space and evaluate their performance based on a specific ML algorithm which attempts to be fitted on the given dataset in order to find the best performing combination of features [51]. Some of these techniques are the forward feature selection, the backward feature elimination and the recursive feature elimination. Lastly, the intrinsic methods use algorithms that include feature importance ranking during their training. These algorithms could be ML models like decision trees, linear models, random forest or regularization methods like LASSO and Ridge regression [52].

For the three classification models presented before, large feature sets were developed and especially for the first model 29 features, to capture the essence of the data. Thus, the feature selection is essential in order to lower the dimensions of the input and at the same time let the models have a better performance. For the feature selection in this project the filter methods are being used, ANOVA and mutual information, because they are faster, less prone to overfitting and can be easily implemented in the cross-validation.

These two techniques have some specific characteristics that differ from each other. ANOVA is used to check if independent groups have means that are statistically different from each other based on the null hypothesis that they are all equal. The F-statistic is used to measure the significance of ANOVA which equals to the ratio of the variance between sample means to the variance within samples [53]. A high F-score corresponds to a feature that discriminates classes well in the sample set. Mutual information is a statistical test that measures the mutual dependence between two random variables, by quantifying how much information can be acquired from a random variable when observing another one variable of the data set [54]. Similar to ANOVA, a high value of the mutual information indicates low uncertainty and hence high dependence of the class on the certain feature.

4.2.2. Hyper-parameter tuning

Hyper-parameter tuning or optimization is called the task performed during the model development where the optimal values for the hyper-parameters of a specific model are chosen. More specifically, for a SVM model with radial basis function kernel, the parameters need to be configured are C and γ. As mentioned in chapter 3, C is the regularization parameter and when is too high, the model penalizes highly the misclassifications, which means that it takes into more account the outliers, so it is overfitting. Parameter γ determines how much a data point influences the general model. When its value is too low, the hyperplane boundary is too general, since it is unable to capture the complexity of the data, so the misclassifications increase. In the opposite, when its value is too large, there is going to be a very complex and curvy hyperplane which leads to overfitting again. Therefore, the values of C and γ is highly related to the existing data set that is going to train the SVM model, and there has to be achieved a balance between bias and variance [55].

There are two ways of searching for the hyper-parameters' values. One is the grid search where the search space is defined with exact values forming a grid. The other way is the random search where the search space is simply a bounded space with sample points randomly selected from it. The former includes exhaustive search, however there has to be a good choice over the search points. The later is only affected by the boundaries of the space so it can lead to better solutions than the former one, but it is more time consuming. In this project, a grid search is used, since there is already some essence captured for the search space from the values being used by Ranganathan [2]. The values inspected during the grid search for C are 1, 10, 100, 1000, 10000 and for γ are 0.0001, 0.001, 0.01, 0.1, 1.

4.2.3. Cross-validation

Cross-validation provides the ability to train and validate the model in separate snippets the whole data set, whereas in a static train-validate set split only a small proportion of the data set is used for the validation. Subsequently, the evaluation on the different validation sets makes the training of the model more reliable and efficient as well, compared to the static train-validate set split. In particular, a k-fold cross-validation splits the current dataset into k equal sets and every time $k-1$ sets constitute the training set and the rest of it the validation set. In this project, a nested cross-validation is being used which includes an outer and an inner cross-validation with $k = 5$. The outer split aims to test the model selected each time with a different

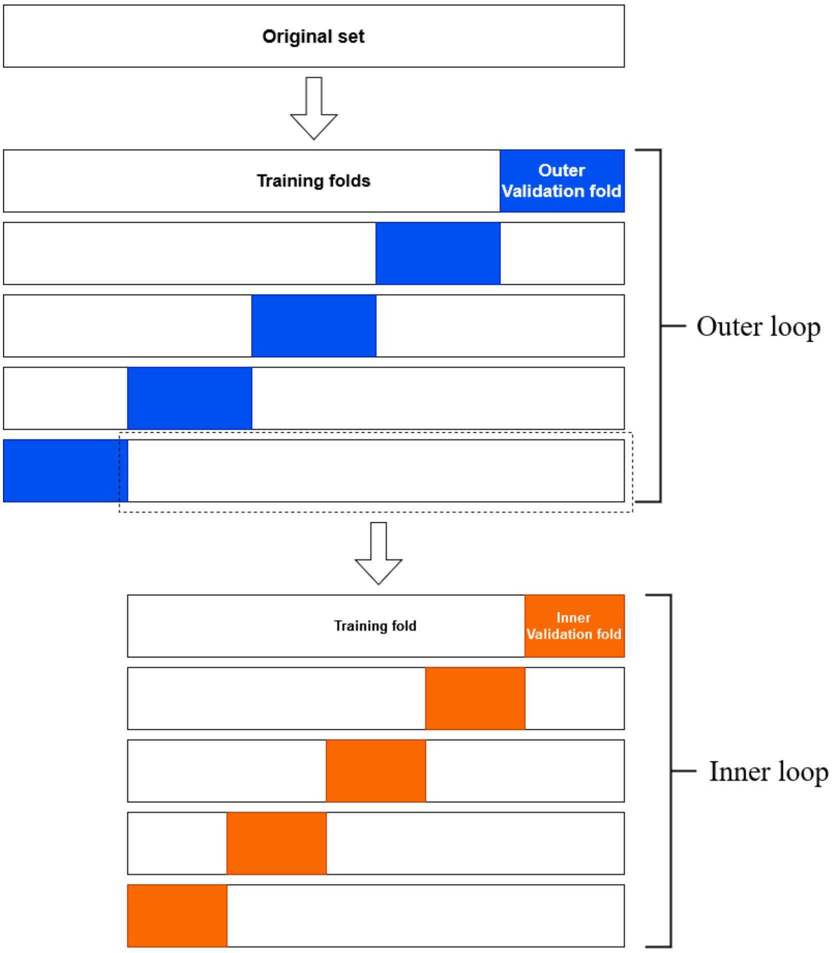


Figure 4.5: Nested 5-fold cross-validation diagram

test set and is responsible for the final evaluation. Hence excluding possibilities of a over- or under-performance due to the specific set conditions or leaking information. The inner split is where the feature and hyper-parameter selection is done, with the grid search. So, at that point the validation set is very useful into checking the accuracy of each set of parameters, rank them accordingly and then select the best ones. In figure 4.5 the diagram of the nested 5-fold cross-validation is shown.

4.3. Results

In this section the model development process for each model is presented, where the steps are all the same for every one. The final macro f1-score of each model derives from the average for each test fold. The hyperparameters' values come from the closest value of the grid search to the average value for each test fold. In a similar way, the feature selection number is done in the same way, while the exact features selected are the most frequent selected out of all the folds. For example, features that are selected in all the five folds are ranked first, then the ones that are selected in four out of five folds and so on. Thus, from this ranking the suitable number of features is chosen as said above. The two models based on the different feature selection methods are presented and following that a final comparison is done amongst the whole models. The original set here is the data set excluding the final test set which is being used in the next chapter for the final evaluation in order to ensure that there is no data leakage to that test set. The cross-validation ensures that all the models are validated in the whole set through different folds. The results shown in this chapter concern the classification model regarding only the stable faults, consists of the first part of the model development for this thesis and it could be the basis for the development of the classification model including the unstable faults in the chapter 5.

4.3.1. Results for the Model 1 with Features in Order

The results for the model 1 can be seen in table 4.6. For both the feature selection methods the model was trained at the same training and validation data sets. The final mean macro f1-score is quite close for both the feature selection methods, whereas the average value of the hyper-parameter γ and the number of features selected is exactly the same. However, the rank of the features is not the same at these two occasions. The most important features with both the ANOVA and the mutual information is $I\_wft\_mean\_3$ and $V\_wft\_mean\_1$. These features are quite differentiating components, because they are crucial for distinguishing a two-phase from a three-phase fault. After those, V_0_mean is essential for the ANOVA method only and $I\_wft\_comparison$ the mutual information method only. The latter can provide straightforward results regarding the fault type. Then features coming from the sequence components are significant for both the feature selection methods such as I_1_mean, I_2_mean, I_2_mean/I_1_mean showing that they are significant for the fault classification and features directly from the windowed FT such as $I\_wft\_mean\_2$

Table 4.6: Results for model with features in order

Method	ANOVA	Mutual Info
f1-score	0.976	0.973
C	100	1000
γ	0.01	0.01
features	11	11
feature rank	I_wft_mean_3 V_wft_mean_1 V_0_mean I_1_mean I_2_mean I_2_mean/I_1_mean I_wft_mean_2 V_wft_mean_2 V_2_mean I_wft_mean_1 I_wft_harm_energy_3	I_wft_mean_3 V_wft_mean_1 I_wft_comparison I_1_mean I_2_mean I_2_mean/I_1_mean I_wft_mean_2 V_wft_mean_2 V_0_mean V_2_mean I_wft_harm_energy_2

and $V\_wft\_mean\_2$. In the end, both have a selection of a feature based on the third harmonic.

4.3.2. Results for the Model 2 with Per-Phase Features

The results for the model 2 can be seen in table 4.7. For both the feature selection methods the model was trained at the same training and validation data sets. Here there are the exactly same results regarding the average of macro f1-score. The average of the hyper-parameters' values differ on both C and γ. More particular, C with the ANOVA technique is on the upper boundary and γ with the mutual information technique is on the lower boundary. This is not a good indication for the generalization of the model and both can lead to overfitting. The number of features is less with the mutual information and all the features selected in with that, they are selected also by ANOVA with different rank though. The highest importance features selected by ANOVA concern first the current and then the voltage windowed FT. More specifically, the maximum and average value of the current windowed FT and the minimum and average value of the voltage windowed FT. However with the mutual information the mean value of voltage windowed FT, then the minimum impedance and finally the peak-peak current value. Although this method leads to a selection of 3 features, which in general increase the generalization of a model, there is a concern whether this number is too low for the model for learning sufficient information from only these features.

Table 4.7: Results for model with per-phase features

Method	ANOVA	Mutual Info
f1-score	0.984	0.984
C	10000	1000
γ	0.001	1
features	7	3
feature rank	I_wft_max I_wft_mean V_wft_min V_wft_mean I_peak_peak Z_mean Z_min	V_wft_mean Z_min I_peak_peak

4.3.3. Results for the Model 3 with Symmetrical Features

The results for the model 3 can be seen in table 4.8. As before, both the feature selection methods the model are trained at the same training and validation data sets. Here for both the feature selection techniques the results regarding the average macro f1-score are the same. The hyperparameter γ is the same however hyperparameter C is greater for the mutual information. The number of features selected is less with the ANOVA. All the features selected based on the ANOVA, they were selected also by mutual information but with different ranking. The only additional feature selected by mutual information technique is $I\_wft\_comparison$ which is also the selected as the most important one. This can provide a straightforward value that corresponds to each different class of fault. On the other hand, P_ft_mean is the most significant feature based on ANOVA, however it is also ranked second by mutual information. I_2_mean/I_1_mean and then I_0_mean/I_1_mean are the most important features for ANOVA, whereas I_1_mean and then V_2_mean are the following most important features for mutual information. The former set can provide a value between 0 and 1 and based on the combination of those could be a good differentiation factor. In general, features derived from the sequence components can be more interpretable since they are already used in the current practices. Given that the positive sequence voltage and current are available, they provide the basis and a magnitude to compare with the other variables.

Table 4.8: Results for model with symmetrical features

Method	ANOVA	Mutual Info
f1-score	0.980	0.980
C	10	100
γ	0.1	0.1
features	8	9
feature rank	P_ft_mean I_2_mean/I_1_mean I_0_mean/I_1_mean V_0_mean I_2_mean I_0_mean V_1_mean I_1_mean	I_wft_comparison P_ft_mean I_1_mean V_2_mean I_2_mean I_2_mean/I_1_mean I_0_mean/I_1_mean V_0_mean I_0_mean

4.3.4. Comparison of the three models

The review of the final models selected is given in the table 4.9. First the model with the highest macro f1-score is the one with per-phase features both with ANOVA and mutual information. Second, for each model macro f1-score with the two different feature selection techniques are the same or quite close. More precisely, only for the model with the sorted features the feature selection techniques do not end up to the exactly same result, but a bit different. Third, only the per-phase model with the mutual information leads to a significant lower number of features selected compared to ANOVA. Fourth, the sorted features model seems to lack in performance compared to the other two models by around 5-10% lower average macro f1-score value from the other models. This is not exclusively the only reason of selecting a model, however at those high percentages this is an essential drawback to take into consideration. Finally, the hyperparameter's C values differ in every case, as values from 10 to 10000 are chosen and for γ the values chosen are 1, 0.1, 0.01 and 0.001. The ones where $C = 10000$ and $\gamma = 1$ are the most concerning, namely the per-phase features with ANOVA and mutual information, as these consist of the upper and lower boundary of the search space, respectively. After investigating the behaviour of the first case when the upper limit of the search space is extended to 100000, for some folds indeed the new limit is selected. This implies a peculiar behaviour of the training model in selecting the highest values of C that could lead to overfitting and subsequently not a good performance with unseen data. Similar response there is for the other model when the search space for γ includes the value 10, which leads to the same considerations regarding the generelization of such a model. This model also selects 3 features, which brings doubts regarding the ability of acquiring the sufficient internal information of the model based only on a few features.

Table 4.9: Final models

	Sorted features		**Per-phase features**		**Symmetrical features**	
Method	*ANOVA*	*Mutual Info*	*ANOVA*	*Mutual Info*	*ANOVA*	*Mutual Info*
f1-score	0.976	0.973	0.984	0.984	0.980	0.980
C	100	1000	10000	1000	10	100
γ	0.01	0.01	0.001	1	0.1	0.1
features number	11	11	7	3	8	9

Moreover, for the sorted features model, the way of decoupling the phase information is forced based on the values ranking which is not the purest way for for making the model phase invariant. For example, using minimum, maximum and medium values, the medium one loses the sense as what reflects on. It could be the medium, but sometimes is closer to the minimum and sometimes to the maximum. On the other hand, the features of symmetrical model are developed that way so there is a natural phase symmetry, assuring the model is symmetrical under permutation of the phases A->B->C->A. Also, as mentioned before in the section 4.1.1, for the per-phase model there is need for the historical data to be suitably labeled with exact phase information regarding the fault state. Since Alliander currently does not use such labels, for the purpose of this model the labels were created manually. However for practical reasons this requirement for labeling, halts the use of this specific model. Furthermore, an argument in favour of the symmetrical features model is that the features used are more explainable compared to the others, since their values can indicate better the derivation of the result. The symmetrical components are already being used by Alliander for the current classification model, so there is more familiarity with these components and can be more easily interpreted and trusted.

All these arguments lead to the selection of the third model and more specifically the one with ANOVA feature selection technique among the six developed models, since the selection of less features is a good indication for generalization, while the lack in performance is not that significant. Morevover, among the other models the symmetrical features model can provide more explainable features. Therefore, under the umbrella of stable faults, this model could replace the current fault classification model. The advantage over the current Alliander classification is that the procedure can exploit the huge amount of data stored in the database by the SASensors which provides potential of further improvement, since the data only keeps on increasing. The Alliander classification model for the stable faults (single-, two- and three-phase) is still more accurate, with 8 misclassifications over the whole dataset for the stable faults given consisted of 315 entries. The performance of Alliander's algorithm corresponds to macro f1-score around 0.986 for the three classes of stable faults which is better than the model selected here. So the focus must be in a further development of the classification model here, in order to additionally include the unstable faults and then evaluate the entire performance. This is discussed in the chapter 5. Compared to the model developed by Ranganathan [2], the performance seems to be similar, however the model developed in this chapter is more explainable and subsequently more trustworthy for Alliander to use in practice. The conclusion here is that this model cannot yet effectively fully replace the current fault classification alone, since the inclusion of unstable faults need to be further investigated. However, this explainable fault classification model is a solid foundation of the extended model for all the types of fault, developed and presented in chapter 5.

4.4. Summary

In this section, the development of the three different classification models with explainable features for stable faults and two different feature selection techniques was discussed. The motivation behind the analysis shown before is to develop and select a simple and explainable SVM model based on the historical data existed which can classify future stable faults into single-, two- and three-phase faults with high accuracy. The major achievement of this chapters work is having the basis needed for the extension of this fault classification model for only stable faults, into a fault classification and location model that can handle all the types of fault, as will be shown in chapter 5.

Essential for this project was the explainability which can derive from the features used. Thus this model can help in interpreting the classifier's results, due to the specific ML model and the features used. The features were carefully created that way, so the result could be understandable by Alliander, by exploring their values. Finally, based on the analysis presented in this chapter, a part of another research question could be answered.

What are the methods to evaluate the machine learning model's performance on classifying the faults? The metric used for the evaluation of the ML classification models' performance is the macro f1-score. It is a trade-off between precision and recall, so it can represent the entire performance of the classifier. The use of a nested cross-validation, makes the evaluation more objective as the model can be validated in different sets of the whole dataset and not only in a single split that cannot lead to proper insights. For each of the six models the final score derives from the average value of the five macro f1-scores for each fold and based on that they seem to perform quite well in classifying types of stable faults.

5

Development of Model for Unstable Faults

The goal here is to refine the current Alliander stability thresholds and explore ways the current practice can be improved. In this proposed new method the initially labeled unstable faults which have at least one stable period and can provide a reliable fault loop-impedance, will pass through the automatic model. An estimation of the fault location could be available directly to the control room and thus the isolation of the faulty section could be faster. Subsequently, the interruption of the power supply is less which is a result that the operators aim for.

5.1. Model Development

As mentioned before, the real faults differ from the ideal ones, both from stability perspective as well as from the presence of more than one type of faults in a fault signal. The stability concerns the waveform of the signal and how close it is to the pure sine waveform. This influences the calculation of the loop-impedance/-reactance in the end, since this is based on the fundamental components and thus there has to be a certainty about the fault waveform. It is essential to end up in a reliable loop-impedance/-reactance so subsequently the derived location estimation is as accurate as possible. The current algorithm of Alliander suggests that if the thresholds given in table 2.2 are met, then the fault is classified as unstable and no loop-impedance/-reactance is sent to the control room. However, after manual inspection by experts some faults labeled as unstable, are in reality stable faults with a location that could be derived also from the loop-impedance/-reactance. This indicates that the current stability thresholds are conservative and a refinement of them is required, so the most serious stable part of a fault is recognized. That way the calculation of the loop-impedance/-reactance could be reliable, as it depends on the fault type. In general, the majority of unstable faults are single-phase faults, so using zero sequence components would be appropriate for that. However, for the waveforms with more than one fault types, two-phase and three-phase faults have to be included and since the zero-sequence is not predominant in these types, different ways need to be explored. So, for the case of a multi-phase fault, the phase-phase signals are used. That way the effect of the zero-sequence component distortions are canceled out and the check for instability is more reliable.

Due to the existence of different type of faults in one single entry and the need to develop a model that includes the unstable faults, the segmentation algorithm here needs to be altered. In particular, the multivariate input segmentation method is being used which was described in chapter 3 with three break points. So, two segments can be formed, one starting from the first point to the second point and one from the second point to the third one. If each of these parts consists of one period or none, then the result is considered to only consist of one part. The model developed and selected for the stable faults in chapter 5 needs to be adapted and retrained. Thus, it is used as a basis for the more general model developed in this chapter in order to encompass all the faults including the unstable ones. The new model developed has broader range, since it does not consist only of the ML model. Figure 5.1 shows this new model. In the rest of this section the model is further analysed.

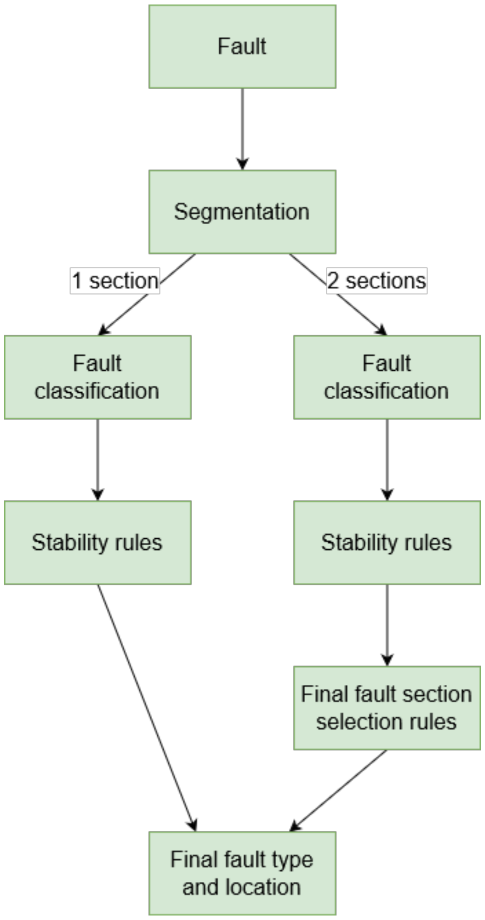


Figure 5.1: Updated parts on the current restoration process

First of all, the ML model now provides intermediate results regarding the fault type of each segmented part and not the final result. For each signal one or two segmented parts enter the ML model where the output is the type of fault for each part separately. The fault type could be single-phase, two-phase, three-phase or self-extinguishing fault. For this reason, a manual labeling was performed for each segmented period in order for the training and testing of the ML model. The main problem of the model developed in chapter 4 and in [2] is that event recordings that consist of more than one fault types, normally detect the most serious type, however there is no alternative on that if the stability rules do not hold. For example, the signal given in figure 5.2 is labeled as a single-phase fault. Visually, it seems like a two phase fault, however that part of the signal is unstable. So, in order to find a reliable fault loop-impedance/-reactance, the single-phase part is checked and indeed it can be characterised as stable. The way the models of chapter 4 and [2] are developed, this fault would be considered as two-phase fault or as fully unstable fault. Hence the development of the model in this chapter it could be considered as an acceptable procedure since the target values were only these four specific categories, with the unstable label to come after the stability rules model, as it is being explained below.

Following that, for each part the set of stability thresholds is being checked period per period, except for the segments that are labeled as self-extinguishing faults. Regarding the rest of the types, if there is at least one stable period along the segmented part for the permanent faults then the fault is considered stable and the type of this part remains the same as from the ML model. Otherwise the part is labeled as an unstable fault.

Next the final decision over the whole signal has to be done. When there is only one segmented part then the decision is simply the final label of that part after the stability check. When there are two segmented parts then the decision is a bit more complicated, so there are some rules on that. The table 5.1 provides these

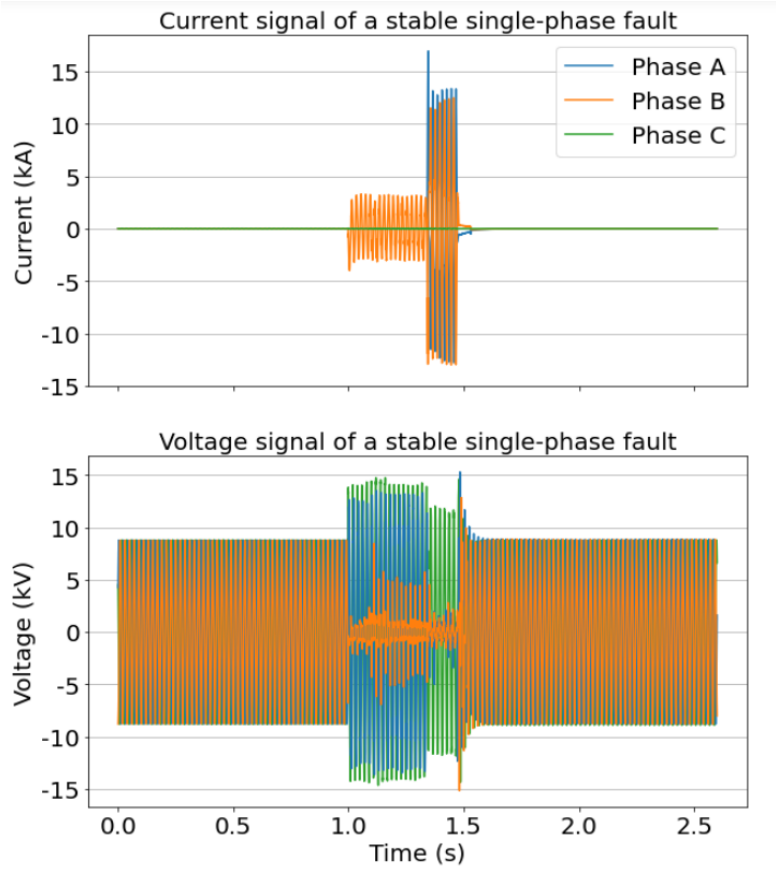


Figure 5.2: Example of a stable single-phase fault that also has unstable two-phase fault part

rules. In that table, the label self-extinguishing indicates the possibility of a self-extinguishing fault at a specific segment, as such training labels were used also for segments with not any of the other stable fault labels. Basically, as said before the most serious stable fault is promoted even when the other part is a suspected self-extinguishing or unstable fault. When there is one suspected self-extinguishing and one unstable fault, the unstable is the final result. The unstable fault is more serious than the self-extinguishing fault, since the power supply is disrupted and a restoration need to be done. Self-extinguishing fault does not require a direct fault location, that's why is not promoted over unstable fault. When both parts have the same type of fault, then the final fault is going to be the same, but the exact part selected is the one that provides the lowest fault loop-impedance/-reactance.

The attempt to find the final fault type of a signal aims to the further estimation of fault location. As said before, a possible fault location is going to help the quicker fault section isolation and restoration. The loop-impedance/-reactance is calculated for each stable period using the correct equation from the table 2.3 depending on the fault type. Then for each part the fault loop-impedance/-reactance has to be selected. The concept is similar to the one proposed by Gu and Provoost [3] and in particular, consists of the minimum value of the loop-impedance/-reactance out of the whole stable part. That way the possibility of calculating a wrong impedance due to the transients at the end of the fault, is diminishing. Each part has now the loop-impedance/-reactance calculated. If there is only one segmented part, then this is the one that leads to the final fault loop-impedance/-reactance. If there are two segmented parts, then the choice of the part is done as explained before based on table 5.1 and if the faults are the same, the lowest between the two loop-impedances/-reactances is the final one.

The difference with the current algorithm that Alliander uses, is that the fault classification is done with ML based technique whereas Alliander uses the criteria shown in table 2.1 and the stability thresholds are less conservative. These new stability rules are developed and shown in the section 5.2. The fault classification is done with a more automatic way taking advantage of the historic data stored by the SASensor and since the

Table 5.1: Set of rules for the final fault type when there are two segments

First part	Second part	Final result
single-phase	single-phase	single-phase
single-phase	two-phase	two-phase
single-phase	three-phase	three-phase
single-phase	self-extinguishing	single-phase
single-phase	unstable	single-phase
two-phase	two-phase	two-phase
two-phase	three-phase	three-phase
two-phase	self-extinguishing	two-phase
two-phase	unstable	two-phase
three-phase	three-phase	three-phase
three-phase	self-extinguishing	three-phase
three-phase	unstable	three-phase
self-extinguishing	self-extinguishing	self-extinguishing
self-extinguishing	unstable	unstable
unstable	unstable	unstable

data recordings only are increasing, such a fault classification model has the potential to get even better in the future. For the stability part, a per-period analysis is still being used since the importance of that task's result does not let a lot of space to deviate from that. More precisely, if even one well defined stable period can provide a reliable fault loop-impedance/-reactance calculation, it benefits the estimation of the fault location. As the target is to find as more as possible these kind of faults that otherwise would be labeled as unstable, makes exploration of each single period essential. Figure 5.3 depicts with yellow color the part of the current procedure that is altered with the new suggested model.

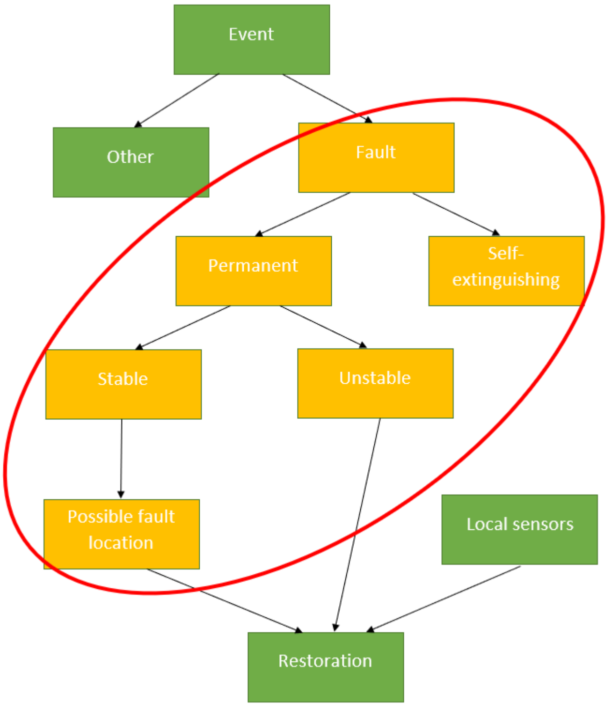


Figure 5.3: Updated parts on the current restoration process

5.2. Development of Features for Stability

5.2.1. Feature Development

The stability classification of a signal comes from the waveform. The current and voltage waveforms ideally should be pure sinewaves, without any distortion. This means that the Fourier transform (FT) should contain no harmonic components. The fundamental component of the FT of the current and the voltage is used for shaping the symmetrical components and the impedance/reactance. Then, it is significant to make sure that the waveforms are pure enough, so they provide a reliable loop-impedance/-reactance which subsequently leads to an accurate fault location. This directs the attention for developing features that can measure the "distance" between the real signal and its corresponding ideal sinewave. Consequently, the issue is how to create an ideal sinewave which corresponds to the real signal. This is done in two ways so there is a compliment of the ideality. One way is by taking the fundamental component of the FT of the signal and then transform it back to the time as a sine. The other way is by applying a median filter with 7-points window to the real signal, so to smoothen out sudden momentary huge distortions like peaks due to restrikes and transients.

Regarding the first metric, the real signal is compared with the corresponding ideal one point-point. This ideal signal corresponds to the inverse FT of the real signal where only the fundamental frequency component exists. In particular for each period that consists of $N_s = 80$ points, the root mean square error $rmse$ is calculated as such:

$$rmse = \sqrt{\frac{1}{N_s}\sum_{n=1}^{N_s}(real_i - ideal_i)^2} \tag{5.1}$$

Then, this magnitude in order to be scaled down to per unit, so there is a uniform comparison, is divided by the maximum value of that same period for the ideal signal ($\max ideal$). Thus, the metric $rmse\_max$ is formed as such:

$$rmse\_max = \frac{rmse}{\max ideal} \tag{5.2}$$

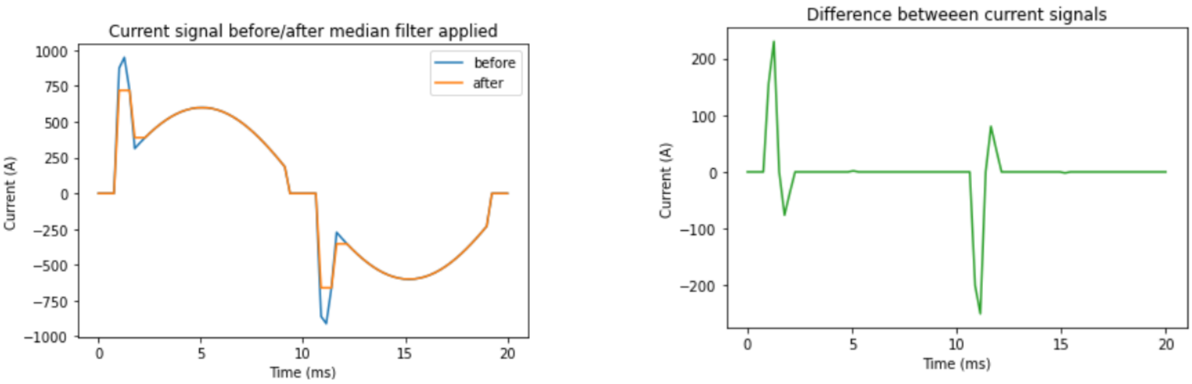
This value is expected to be close to zero for the stable faults, as such a signal should have normally no difference to the corresponding pure sinusoidal signal. This metric is used for the zero-sequence and phase-phase current and voltage signal.

The metric where the median filter is applied is shaped as such:

$$ratio = \frac{medfilter\_ft\_fundamental}{real\_ft\_fundamental} \tag{5.3}$$

More precisely, for this criterion first the median filter with 7-points window is applied to the real signal. That corresponds to the replacement of each sample point's value by the median value derived by the 7-points window around that sample point, so the values of 3 points before, the current value and the values of 3 points after that. Then the magnitude of the fundamental frequency component of that new signal is divided by the magnitude of the fundamental frequency component of the real signal for each period. The anticipated value of a stable period is around 1 per unit, since the median filter should make only slight difference in a sinusoidal waveform without huge sudden increase or decrease of their value. Subsequently the use of the sliding window FT for calculating the impedances or the symmetrical components it is more reliable and trustworthy. This metric is also applied in the zero-sequence and phase-phase current and voltage signal.

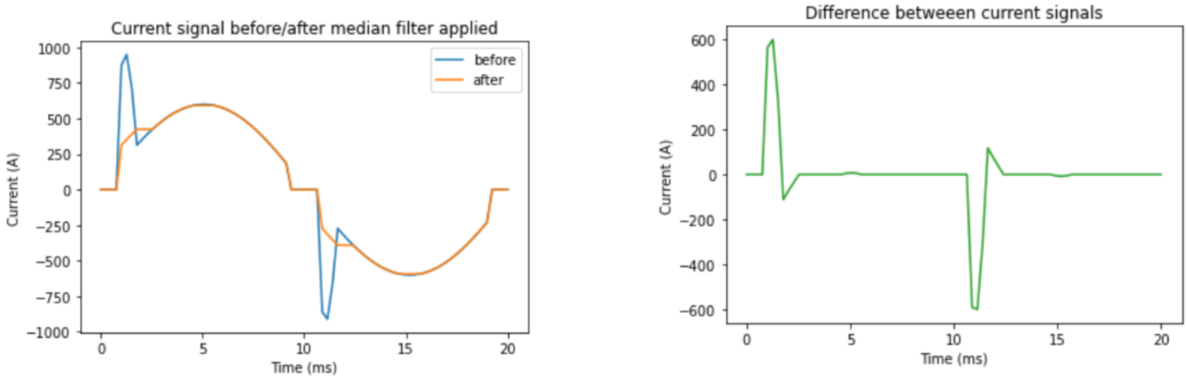
Figures 5.4 and 5.5 depict the differences over a current signal in one period before and after applying a 5-point and a 7-point median filter, respectively, so to compare them. Visually, it seems that the 7-point median filter brings the initial signal in a closest form to an ideal sine wave. Hence, the difference between the signals before and after the application of the 7-point median filter is greater than the one of the 5-point median filter. Subsequently, the 7-point median filter penalizes more the existence of disturbances and thus the instability. The resulted FT for that window regarding the real current signal has fundamental magnitude $633A$ and angle -81 deg. Regarding the real signal after applying the 7-point median filter has fundamental magnitude $607A$ and angle -87 deg. Regarding the real signal after applying the 5-point median filter has fundamental magnitude 627A and angle −83 deg. So, the 7-point median filter leads to a value for the feature $ratio$ equal to 606.73/632.75 = 0.958, whereas the 5-point median filter leads to a value for $ratio$ equal to 627/633 = 0.991. This should correspond to an unstable period of fault, thus a value in a longer distance from $ratio = 1$ would be more appropriate and in particular the one of the 7-point median filter. For comparison, the FT for that window regarding the real current signal without the two peaks is calculated and results to a fundamental magnitude 594A and an angle −89 deg. This indicates that applying the median filter of 7 points to the real signal of this case, a signal which is close to the more ideal form of that particular signal is reached than when the 5-point median filter is applied.



(a) Current signal before and after a 5-point median filter is applied

(b) Difference between the current signal before and after a 5-point median filter is applied

Figure 5.4: Current signal with some disturbances and 5-point median filter applied

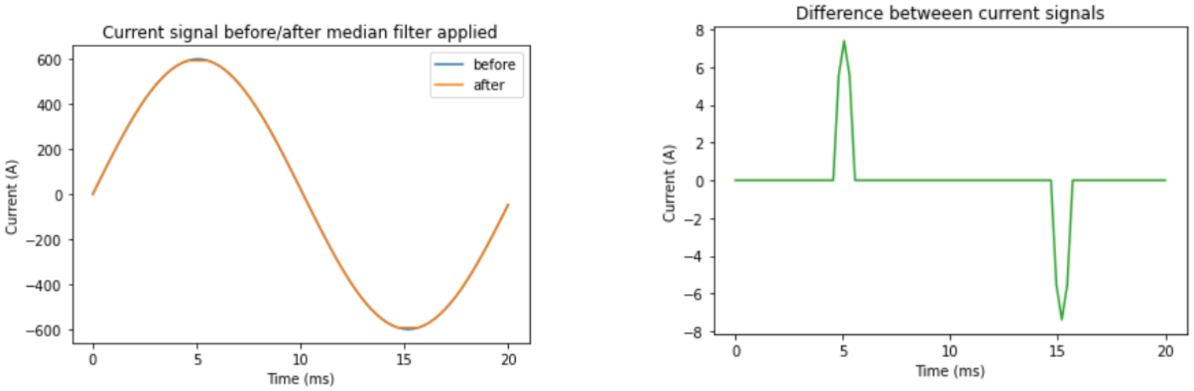


(a) Current signal before and after a 7-point median filter is applied

(b) Difference between the current signal before and after a 7-point median filter is applied

Figure 5.5: Current signal with some disturbances and 7-point median filter applied

On the other hand, applying a 7-point median filter to a pure sinusoidal signal would have minimal impact on its waveform and subsequently to the FT magnitude and angle derived. In particular, for the current signal shown in figure 5.6, there is a small difference at the two peaks and the FT fundamental magnitude deviates a bit between the real signal and the signal after the median filter, with values 600A and 599A, respectively. The resulted value for the feature $ratio$ in that case is 599/600 = 0.998 and it should indicate a stable period of fault. A 5-point and lower median filter would have in that occasion same or better response, but it is discarded due to the previously discussed behaviour. A 9-point or greater median filter would flatten even more the peaks, that's why the final choice is the 7-point median filter.



(a) Current signal before and after a 7-point median filter is applied

(b) Difference between the current signal before and after a 7-point median filter is applied

Figure 5.6: Ideal current signal and 7-point median filter applied

5.2.2. Feature Evaluation

The goal with the refinement of the stability metrics is to correctly classify stable faults that would be characterised as unstable faults based on the Alliander rules. The most trustworthy element that could be used as evaluation is the final loop-impedance/-reactance. In the historic data provided, the fault loop-impedance is given for each data entry calculated based on the result of the Alliander classification algorithm. This can be noted as the real fault loop-impedance since the resulted location from this is verified as an acceptable one. The reason of this is that for the unstable faults that afterwards they were identified as stable ones based on the offline inspection, it was noticed that the loop-impedance calculated by Alliander FaultFinder initially, could lead to accurate results. So it can be used as the real fault loop-impedance as it provides a very good indication for the evaluation. Therefore comparing the result of the model developed with the real fault loop-impedance can help with evaluating the stability thresholds eventually. Ideal situation would be to have as much as possible correctly classified with fault loop-impedance close to the real one. At the same time, faults which the fault loop-impedance deviates more than the acceptable limits they should be classified as unstable faults. Therefore, the chosen evaluation is going to be held based on how close to the real fault loop-impedance is the calculated one for the faults initially labeled as unstable.

The process of the stability thresholds model tuning and evaluation is done manually based on the unstable faults training set (solid blue box in figure 3.1) as mentioned in chapter 3. In this set there exist some faults that after the offline inspection of their waveform, they were identified as possible stable faults, so there would be an indication at the number of reclassified as stable faults expected. Also, in this set there are some self-extinguishing faults. For the training of the ML model the single-, two-, three-phase and self-extinguishing faults are used (green and orange solid boxes in figure 3.1). To deploy this model with one or two segments for a certain faulty event, a manual labeling is held per segment, based on the real label of the whole event. That means that in the case of one segment existing, the label corresponds to the real one and in the case of two segments existing, at least one of them it should be the same as the real label and the other is assigned manually. As mentioned above, the last label self-extinguishing refers to the self-extinguishing faults as a definition, but also to the faults that do not belong to any of the other three classes. The relaxation of the definition here is an assumption based on the fact that self-extinguishing faults are the least serious class of faults, with no immediate actions needed. The unstable segments and subsequently faults are being derived after the stability check.

First an appropriate ML model is selected and stays fixed. This concerns the fault classification, so then the suitable stability rules applied since they are fault type dependant. For each fault initially labeled as unstable by Alliander, the final fault type and loop-impedance is calculated. If the type is still unstable the value of the impedance is a default one and there is no meaning on comparing it with the real one. The faults which are classified now as single-, two- or three-phase faults though, their predicted loop-impedance need to be evaluated. The metrics preferred for the evaluation of the results are root mean square error (RMSE), mean absolute error (MAE) and the 90th percentile of the absolute error as explained in chapter 3 and the number of initially unstable faults of the training data set reclassified as stable faults.

The evaluation of these criteria and their values is not a straightforward procedure. Since these values concern per period characteristic, there are no available labels for each one of those, that would allow a supervised ML model to be applied. The manual labeling is a very time-consuming process with around 200 periods at minimum per signal which also would insert bias to the model, as there are no clear waveform stability margins in the theory. The unsupervised ML and in particular clustering, at first glance may seem like an suitable solution. There is no requirement for target values and that makes the building of such a model quite easier. However, the thresholds to be defined here have fine margins, such as 10^{-2} and a clustering method seems in the end to be not appropriate as well. The way it works, creating clusters based on particular characteristics is very good for the big picture, but whenever comes to the exact values that shape the margins, it is not simple to trust that. Certain formulation of the way the distances are taken and the threshold of those distances need to be done in order to have safer conclusions. Since even that does not assure either totally the results of that, the clustering is not used for the evaluation of the stability criteria.

Therefore, the training of the stability rules model is the one noted next. Since $rmse$ and $ratio$ ideally would have values 0 and 1 respectively, there must be a search space around those values. After exploring the respective $rmse$ values of stable faults, the search space for both current and voltage should be between 0.2 and 0.5. At the same manner, $ratio$ values for both voltage and current had to be in the window [0.9, 1.1], however the first changes were noticed at 0.95 and 1.01, so in the end the search space shrank down to [0.94, 1.02]. So, the tuning of these values was held by a sensitivity analysis where different combinations of the thresholds' values inside the search space would be applied. Then for each combination a set of error metrics would be calculated. More precisely, the RMSE, MAE and 90th percentile for both the single-phase and multi-phase faults of the validating data set of the reclassified as stable faults and the number of these reclassified stable faults. In the end, the choice of the appropriate set of stability thresholds is going to be based on the requirements existing regarding the error values and the expected number of stable faults. The last one is based on the indication of possible stable fault after manual inspection by an expert from Alliander. In the next section, the results for the stability rules model and their evaluation are further discussed.

5.2.3. Stability Rules Model Results

The results of the ML model used for the stability evaluation are shown in figure 5.2. As explained before first a ML classification model is selected based on a feature selection, hyper-parameter tuning and nested cross-validation using only the data with type single-, two-, three-phase and self-extinguishing. The data for the initial unstable faults are used as validation set, so there could be a evaluation of their results in the end. The symmetrical features model is used, as this was the one selected in chapter 4, based on the explainability and performance. Both feature selection techniques ANOVA and mutual information lead to the same feature set. The need for creating and training another model and not using the exact same one, is because now

Table 5.2: ML classification model used during the stability model training

Model	Symmetrical features
C	1000
γ	0.001
features	7
features selected	I_0_mean I_1_mean I_2_mean V_2_mean V_0_mean/V_1_mean I_2_mean/I_1_mean I_0_mean/I_1_mean

a different segmentation technique is used which changes a bit the nature of data entering the classification model. Also, the classes are increasing since the self-extinguishing faults are included. The unstable faults are included too now, but at the classification model they do not represent a separate class, they are labeled based on the type of the fault among the classes single-, two-, three-phase and self-extinguishing faults accordingly and not as unstable faults explicitly.

Based on that classification model, after that the initially labeled unstable faults are passing through the stability check model so their stability is evaluated. Then the final fault type and location are calculated based on the criteria of the two segmented sections and the minimum loop-impedance/-reactance. For different sets of values for the stability features the RMSE, MAE and 90th percentile of the classified stable faults by the new model and the number of those is calculated. In particular, as said in chapter 3 the impedance/reactance values calculated by Alliander are compared with the corresponding predicted impedance/reactance values by the proposed model. Table 5.3 shows the values used for $rmse$ and $ratio$ for each set and table 5.4 presents the results for each of those. The thresholds are the same for the zero sequence and the phase-phase waveforms. As said before for the $rmse$ it was expected a low value close to zero and for the $ratio$ a value close to 1, hence these specific values for each feature.

Table 5.3: Thresholds for different sets of stability rules for both zero sequence and phase-phase current and voltage waveforms

	Zero sequence and phase-phase current		**Zero sequence and phase-phase voltage**	
	RMSE	**Ratio**	**RMSE**	**Ratio**
1	I_rmse ≤ 0.2	0.94 ≤ I_ratio ≤ 1.02	V_rmse ≤ 0.2	0.94 ≤ V_ratio ≤ 1.02
2	I_rmse ≤ 0.3	0.94 ≤ I_ratio ≤ 1.02	V_rmse ≤ 0.3	0.94 ≤ V_ratio ≤ 1.02
3	I_rmse ≤ 0.3	0.95 ≤ I_ratio ≤ 1.02	V_rmse ≤ 0.3	0.94 ≤ V_ratio ≤ 1.02
4	I_rmse ≤ 0.3	0.94 ≤ I_ratio ≤ 1.02	V_rmse ≤ 0.3	0.95 ≤ V_ratio ≤ 1.02
5	I_rmse ≤ 0.3	0.94 ≤ I_ratio ≤ 1.01	V_rmse ≤ 0.3	0.94 ≤ V_ratio ≤ 1.02
6	I_rmse ≤ 0.3	0.94 ≤ I_ratio ≤ 1.02	V_rmse ≤ 0.3	0.94 ≤ V_ratio ≤ 1.01
7	I_rmse ≤ 0.3	0.94 ≤ I_ratio ≤ 1.02	V_rmse ≤ 0.3	0.95 ≤ V_ratio ≤ 1.01
8	I_rmse ≤ 0.3	0.94 ≤ I_ratio ≤ 1.01	V_rmse ≤ 0.3	0.95 ≤ V_ratio ≤ 1.01
9	I_rmse ≤ 0.3	0.94 ≤ I_ratio ≤ 1.01	V_rmse ≤ 0.3	0.95 ≤ V_ratio ≤ 1.02
10	I_rmse ≤ 0.3	0.95 ≤ I_ratio ≤ 1.01	V_rmse ≤ 0.3	0.95 ≤ V_ratio ≤ 1.02
11	I_rmse ≤ 0.4	0.94 ≤ I_ratio ≤ 1.01	V_rmse ≤ 0.3	0.95 ≤ V_ratio ≤ 1.02
12	I_rmse ≤ 0.4	0.95 ≤ I_ratio ≤ 1.01	V_rmse ≤ 0.3	0.95 ≤ V_ratio ≤ 1.02
13	I_rmse ≤ 0.3	0.94 ≤ I_ratio ≤ 1.01	V_rmse ≤ 0.4	0.95 ≤ V_ratio ≤ 1.02
14	I_rmse ≤ 0.3	0.95 ≤ I_ratio ≤ 1.01	V_rmse ≤ 0.4	0.95 ≤ V_ratio ≤ 1.02
15	I_rmse ≤ 0.3	0.94 ≤ I_ratio ≤ 1.01	V_rmse ≤ 0.5	0.95 ≤ V_ratio ≤ 1.02
16	I_rmse ≤ 0.3	0.95 ≤ I_ratio ≤ 1.01	V_rmse ≤ 0.5	0.95 ≤ V_ratio ≤ 1.02

Ideally the choice of feature set values would be the one with the lowest RMSE, MAE and 90th percentile for both single- and multi-phase faults and the highest number of stable faults detected. However there is no such a combination, so there must be a trade-off among these metrics. Since the multi-phase initially unstable faults are only 5, there are no significant deviations of the error and the value for MAE is lower than the error margin of 0.05Ω. So the focus is on the single-phase faults errors. Set 1 that has the lowest MAE, 90th percentile and the second lowest RMSE, leads to the lowest number of reclassified stable faults. On the other hand, set 15 that leads to the highest number of reclassified stable faults, corresponds to the second highest RMSE, MAE and 90th percentile. Hence the solution should be something in between.

Table 5.4: RMSE, MAE, 90th percentile values and number of stable faults for the different sets of stability thresholds for both zero sequence and phase-phase waveforms

	Single-phase faults			**Multi-phase faults**			**Number of reclassified stable faults**
	RMSE [Ω]	**MAE** [Ω]	**90th percentile** [Ω]	**RMSE** [Ω]	**MAE** [Ω]	**90th percentile** [Ω]	
1	0.553	0.213	0.689	0.013	0.009	0.016	30
2	0.491	0.233	0.754	0.075	0.049	0.106	34
3	0.661	0.296	1.050	0.075	0.049	0.106	33
4	0.491	0.233	0.754	0.075	0.049	0.106	34
5	0.491	0.233	0.754	0.075	0.049	0.106	34
6	0.491	0.233	0.754	0.075	0.049	0.106	34
7	0.491	0.233	0.754	0.075	0.049	0.106	34
8	0.491	0.233	0.754	0.075	0.049	0.106	34
9	0.491	0.233	0.754	0.075	0.049	0.106	34
10	0.661	0.296	1.050	0.075	0.049	0.106	33
11	1.069	0.434	1.321	0.075	0.049	0.106	35
12	0.621	0.283	1.050	0.075	0.049	0.106	33
13	0.891	0.313	0.707	0.075	0.049	0.106	39
14	0.988	0.370	0.906	0.075	0.049	0.106	38
15	2.172	0.750	1.633	0.075	0.049	0.106	47
16	2.227	0.805	2.097	0.075	0.049	0.106	46

As mentioned in chapter 2, it must be taken into consideration that 1Ω is the error limit acceptable for a single-phase fault and 0.05Ω for a multi-phase fault. For the multi-phase faults since there is no earth components involved the corresponding error in kilometers is $0.1\Omega/km$ and leads to error in distance of $500m$. However for the single-phase faults there is not a single value that corresponds to the kilometers, since the cable type influences the result. Usually the value corresponding to the distance in kilometers for single-phase faults is around $0.4-0.7\Omega/km$ depending on the type and the caliber of the cable used which leads to an error in distance around $1.5km$. For both the types of faults, all the sets have MAE within these boundaries. However, for the RMSE and 90th percentile this is not the case. RMSE is more sensitive in outliers' errors and here it is important to asses these situations which lead to high errors. 90th percentile shows the error coming from the validating data set, neglecting the highest 10% that usually represents outliers. In that context, the sets 3, 10, 11, 12, 14, 15 and 16 are not selected since the have either RMSE or 90th percentile value of the single-phase faults that are out of the acceptable boundary. Since the set 13 can lead to the highest number of reclassified stable faults, while having RMSE and 90th percentile lower than 1, this is the final choice. In particular, the RMSE for single-phase faults is 0.891Ω, the MAE is 0.313Ω and the 90th percentile is 0.712Ω. The number of reclassified stable faults with that set is 39, whereas the expected possible stable faults are 33 and that means also some additional faults were found to be stable. Figure 5.7 presents the distribution of the single-phase errors for the different threshold sets, enhancing the argument for the specific threshold set choice. For better visualization, the error distribution of set 3 is not shown as it is the same with the one of set 10 and in the same way the error distribution of sets 4, 5, 6, 7, 8, 9 are not presented since they are the same with the one of set 2. The final results of the stability thresholds were given in table 5.5.

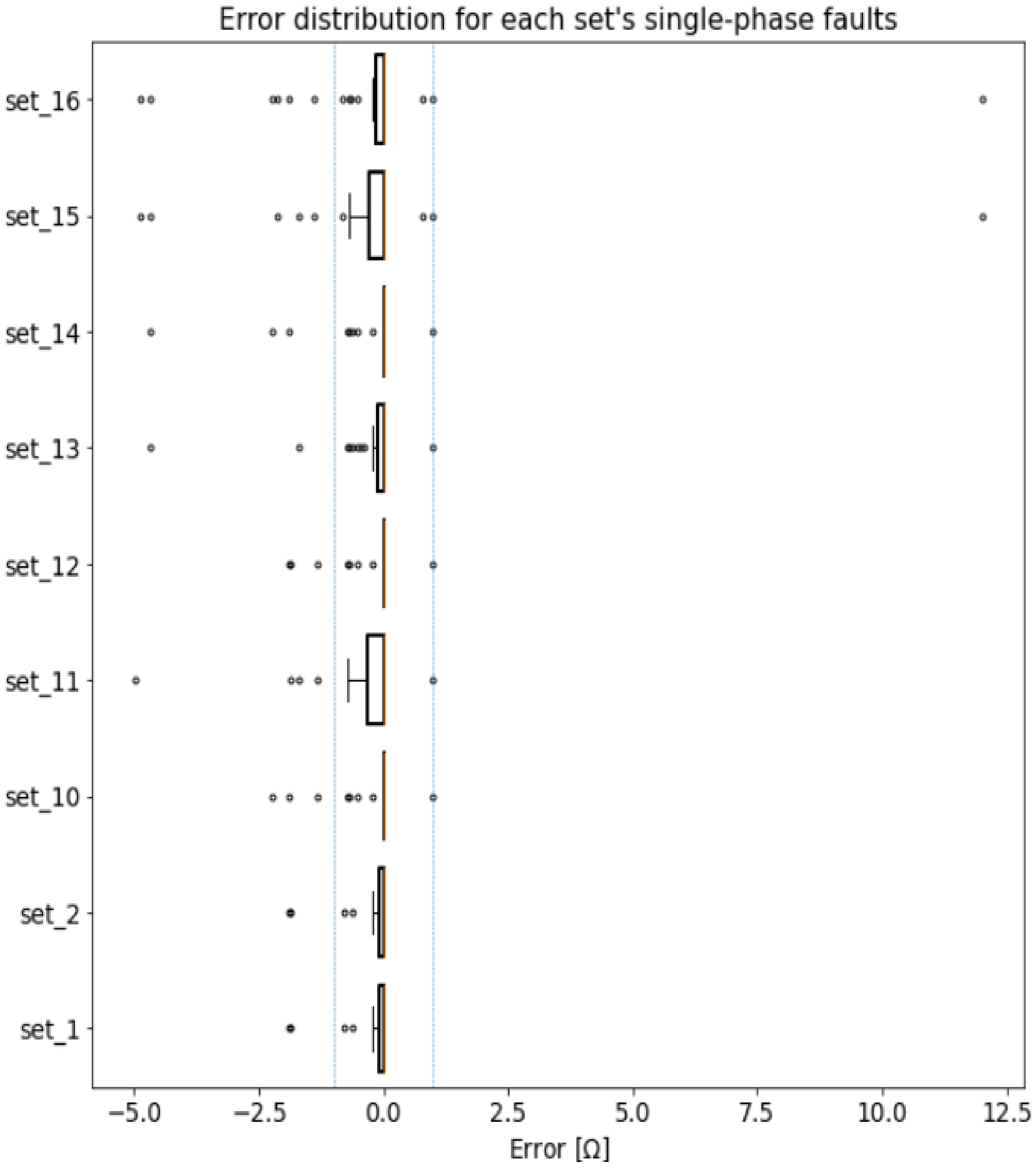
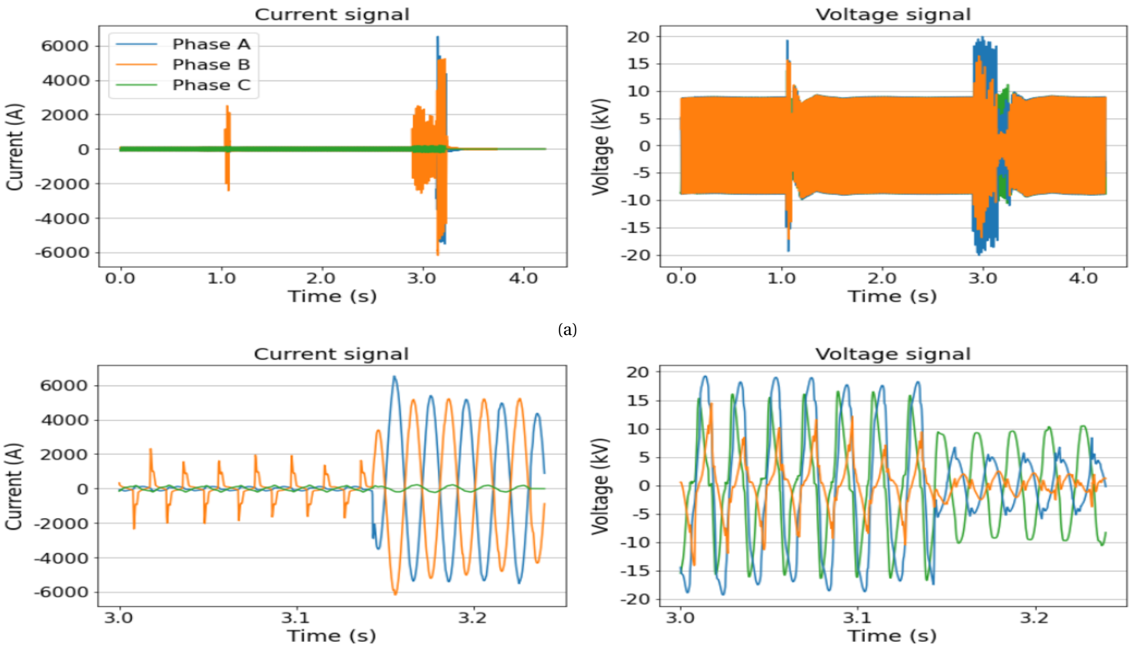


Figure 5.7: Boxplots for the single-phase error distribution of different stability threshold sets

Table 5.5: Stability rules for the updated model

Signal		**Fault type**	
	Single-phase	Two-phase	Three-phase
phase-phase voltage	-	Vpp_rmse <= 0.4 0.95 <= Vpp_ratio <= 1.02	Vpp_rmse <= 0.4 0.95 <= Vpp_ratio <= 1.02
phase-phase current	-	Ipp_rmse <= 0.3 0.94 <= Ipp_ratio <= 1.01	Ipp_rmse <= 0.3 0.94 <= Ipp_ratio <= 1.01
zero-sequence voltage	V0_rmse <= 0.4 0.95 <= V0_ratio <= 1.02	-	-
zero-sequence current	I0_rmse <= 0.3 0.94 <= I0_ratio <= 1.01	-	-

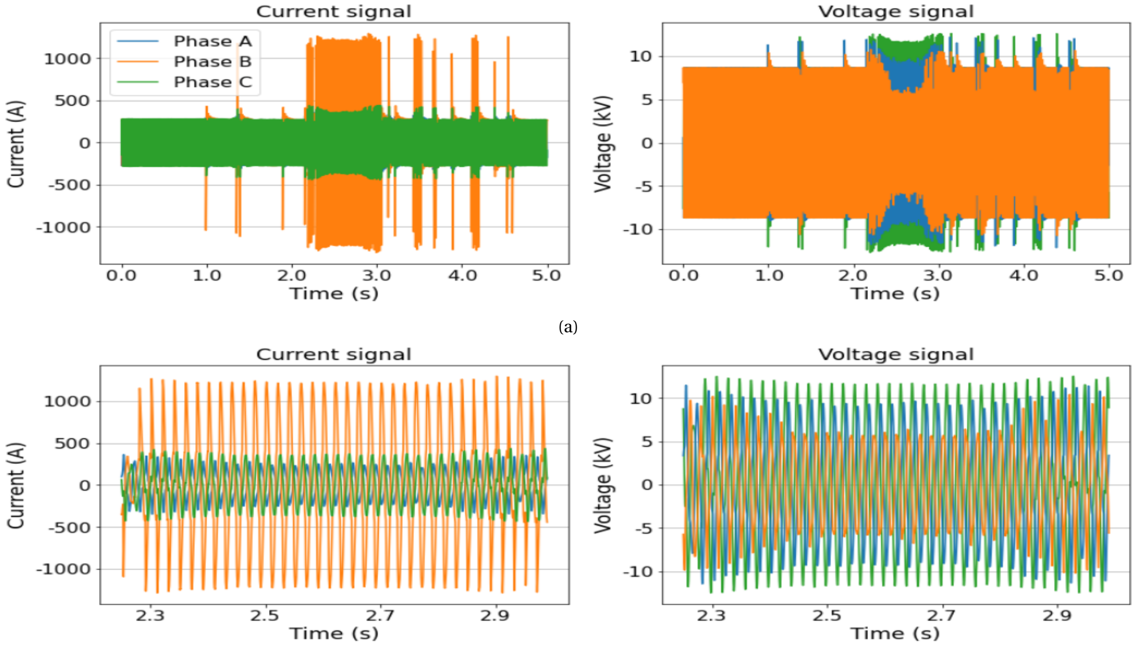
It is worth mentioning that setting these thresholds for the stability model a better stability identification in multi-phase faults can be achieved compared to the Alliander's current algorithm and the model developed by Ranganathan [2]. More precisely, figure 5.8 depicts a two-phase fault that was identified as unstable by Alliander's algorithm and could not pass through the Ranganathan's stability rules because they are only applied to single-phase faults. The stability rules developed in this thesis can correctly identify this two-phase fault and provide an absolutely accurate loop-reactance. Moreover, figure 5.9 shows an example of a fault that has a developed stable part of more than ten periods which was then self-extinguished. This corresponds to a very rare occasion, since normally self-extinguishing faults do not develop a stable part of that many periods. However the location of that fault could be found based on the stable part, for repairing the cable and hence for not letting a future fault to be developed there. This fault was identified as unstable by both Alliander's and Ranganathan's algorithms. Here it is classified as a single-phase fault with absolute error of 0.781Ω regarding the loop-impedance which is clearly below the acceptable boundaries.



(a)

(b)

Figure 5.8: Example of a two-phase fault labeled as unstable by Alliander and correctly classified based on the new stability rules (a) full signal (b) zoomed signal



(a)

(b)

Figure 5.9: Example of an unstable fault labeled as unstable by Alliander and classified as stable single-phase fault based on the new stability rules (a) full signal (b) zoomed signal

5.3. Results of the Entire Model

In this section the classification model selection is presented and then the evaluation of the final results after the stability model are discussed. The classification model is the means of the fault type identification of each partition, before the stability check. After that, the fault loop-impedance/-reactance is calculated and compared to the corresponding real one.

5.3.1. Classification Model Selection

Similar to the procedure in chapter 4, for the proper model selection, feature selection, hyper-parameter tuning and nested cross-validation is used at the same time. The final classification model selected is provided in table 5.6. This final model is trained on the whole training data set (green, orange and blue solid boxes) and evaluated on the test set (green, orange and blue lined boxes) as they are presented in figure 3.1. More precisely, the test set consists of a random 20% of the whole data set regarding the real stable and the self-extinguishing faults plus 6 extra initially unstable faults, 3 initially unknown faults and 8 faults which were misclassified from Alliander's current algorithm. That way there could be a comparison with the performance between the proposed model and the current classification algorithm used by Alliander. These data are hidden from the beginning of the procedure from the first part in chapter 4, even though the data is not entirely the same, so there is no possibility of any information leakage to the model. The values of hyper-parameters come from the closest value of the grid search to the average value for each test fold and the feature selection comes from the most frequent selected features out of all the folds at the average number of features selected every fold. Then the feature set is further decreased by features that are highly correlated with other features, since it was noticed that the testing f1-score afterwards was not affected by that and hence inserting more generalization to the model. Both ANOVA and mutual information are explored during the feature selection procedure, but ANOVA is the technique that leads to the final model selection. The performance score shown corresponds to the average of the macro f1-score for 5 folds on the validation set.

5.3.2. Results of the Model for Fault Classification and Fault Location

Here the model scheme shown in figure 5.1 is evaluated. For the fault classification the model in table 5.6 is selected and is tested on the hidden data. The macro f1-score on the testing set is 0.979 for the sections classi-

Table 5.6: Final ML fault classification model selected

Model	Symmetrical features
F1-score	0.956
C	1000
γ	0.01
features	7
features selected	I_0_mean I_1_mean I_2_mean V_2_mean/V_1_mean V_0_mean/V_1_mean I_2_mean/I_1_mean I_0_mean/I_1_mean

fication and it is a very good result for the identification of each section fault type. Then for the stability rules the values given in table 5.5 are used. Based on those and the selection rules in table 5.1, the final fault type and location are derived. Table 5.7 shows the confusion matrix where the final fault type results in the end are compared with the true fault types after the manual inspection. Table 5.8 depicts the confusion matrix where the fault type results from the Alliander's current algorithm proposed by Gu and Provoost [3] are compared with the true fault types after the manual inspection. Here the faults labeled by this algorithm as unknown are excluded from the table. Table 5.9 consists of the error metrics results for fault location among the stable faults and figure 5.10 shows the distribution of the error for both single- and multi-phase stable faults. From these metrics for the stable faults they are excluded the faults that their given loop-impedance is calculated on single-phase basis but the prediction concerned a loop-reactance calculated based on multi-phase equations and the single-phase faults in which there was no loop impedance available, since they were initially declared as unknown. The full list of the digital fault recording (DFR) file names with the true fault type, the labels from the two algorithms and the result regarding the fault loop-impedance value for each of the two algorithms are given in table B.1 in Appendix B.

Table 5.7: Confusion matrix with the evaluation of the final test set based on the proposed algorithm results

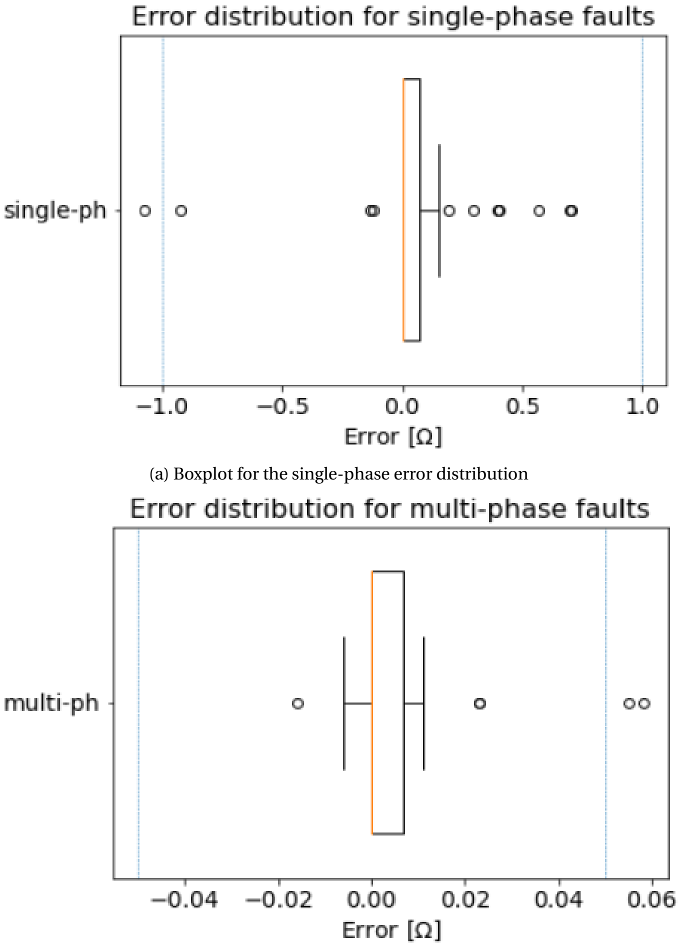
		Predicted value				
		single	two	three	self-ext	unstable
True value	single	33	1	0	0	0
	two	0	24	1	0	0
	three	0	0	15	0	0
	self-ext	3	0	0	83	4
	unstable	0	0	0	0	2

Table 5.8: Confusion matrix with the evaluation of the final test set based on the Alliander's current algorithm results

		Alliander value				
		single	two	three	self-ext	unstable
True value	single	27	0	0	0	4
	two	2	23	0	0	0
	three	0	0	15	0	0
	self-ext	6	0	0	84	0
	unstable	0	0	0	0	2

Table 5.9: RMSE, MAE, 90th percentile values for the final test set

Single-phase faults			Multi-phase faults		
RMSE [Ω]	MAE [Ω]	90th percentile [Ω]	RMSE [Ω]	MAE [Ω]	90th percentile [Ω]
0.322	0.152	0.555	0.015	0.007	0.016



(a) Boxplot for the single-phase error distribution

(b) Boxplot for the multi-phase error distribution

Figure 5.10: Boxplots for the error distributions for the faults in the test set

From the confusion matrix for the predicted values, one could depict a large number of self-extinguishing faults and that's because they outnumbered by a lot all the other type of faults in the possessed dataset. The misclassifications labeled as unstable faults are not serious since this type of faults are a matter of further investigation. The misclassifications labeled as single-phase faults are more serious since a loop-impedance is sent immediately to the control room for depicting the location. However, these may lead to the correct location either way for repairing the cable locally and preventing a future stable fault developing. Regarding the unstable faults, since a huge amount of them was previously used for the training of the stability model, only two were left out and indeed they were correctly classified. Regarding the misclassifications of a single-phase and a two-phase fault, only the former is concerning as the fault loop-impedance given is calculated with the single-phase fault equation. That way the predicted outcome has lower value of reactance and impedance magnitude, as expected since it is a two-phase fault. The latter can be evaluated for the value of loop-impedance that leads to and it is indeed very close to the real one. The values of RMSE, MAE and 90th percentile are indicating that the loop-impedance calculation from the model is quite accurate as they are within the acceptable margins of 1Ω and 0.05Ω for single-phase and multi-phase faults respectively. The boxplots for the error distribution of both single-phase and multi-phase faults points out that the majority of the results is within the desired limits, as well. As a consequence, the location of the fault would be also within the acceptable margins in terms of kilometers, which as mentioned before is around $1.5km$ and $500m$ for single-phase and multi-phase faults, respectively.

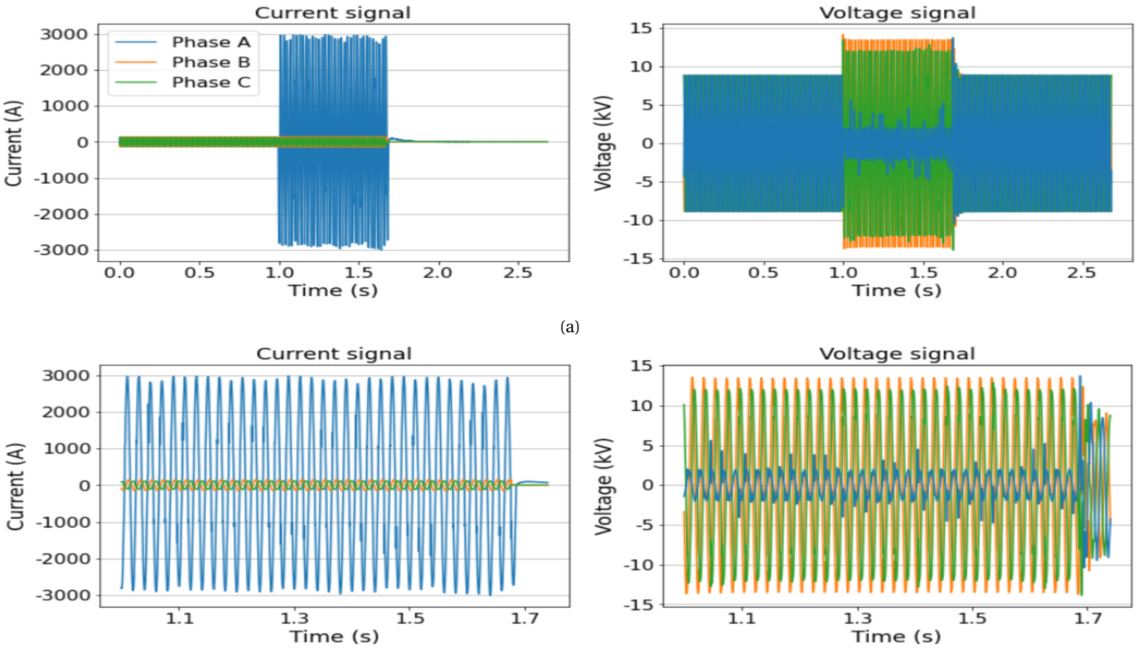
5.3.3. Discussion

The models developed here for the fault type classification and location that include both stable and unstable faults seem to work sufficiently well based on the results in the previous section. As mentioned in chapter 3, with the current Alliander's model only 8 stable faults are misclassified out of the whole data set. To better compare, these 8 stable faults are included in the test set. The proposed model can classify correctly the 2 two-phase faults and the 3 out of the 6 self-extinguishing faults. There can be a comparison of the general performance for the proposed and current model which are shown in the confusion matrices in tables 5.7 and 5.8 correspondingly based on the true fault labels. The proposed model leads to a weighted f1-score of 95.1% whereas the current algorithm 93.3% (excluding the 3 faults labeled as unknown).

The great importance of this model rises into the unstable faults classification. The rules applied in this model, let it correctly classify faults which after the offline inspection it is observed that they can provide a reliable fault loop-impedance and initially were classified as unstable faults by the current Alliander's algorithm. An accurate loop-impedance of these initially unstable faults is sent to the control room and it is going to lead to an accurate fault location. Thus, the locations are provided for more faults, helping into minimizing the total restoration time. This constitutes the best advantage of this model over the current model used by Alliander.

In particular, in the testing data set there exist 6 samples initially labeled as unstable faults by the current algorithm from which 4 are in the end single-phase faults and 2 are still unstable. Moreover, there are 3 entries which were classified as unknown faults by Alliander's algorithm that in reality are single-phase faults. All these faults were classified correctly by the fault classification and location scheme developed in this thesis which as said before leads to the decrease in the overall restoration time. One example for each case is shown below. Figure 5.11 presents a single-phase fault that was labeled unstable by Alliander's algorithm and by the proposed fault classification scheme is classified as single-phase fault. Figure 5.12 shows an unstable fault that was labeled unstable by the current model and it is still classified as unstable. Finally, figure 5.13 depicts a single-phase fault initially labeled as unknown by Alliander and then classified as single-phase fault by the proposed model. In Appendix A the rest of the faults belonging to the test set are plotted that initially were labeled as unstable or unknown faults.

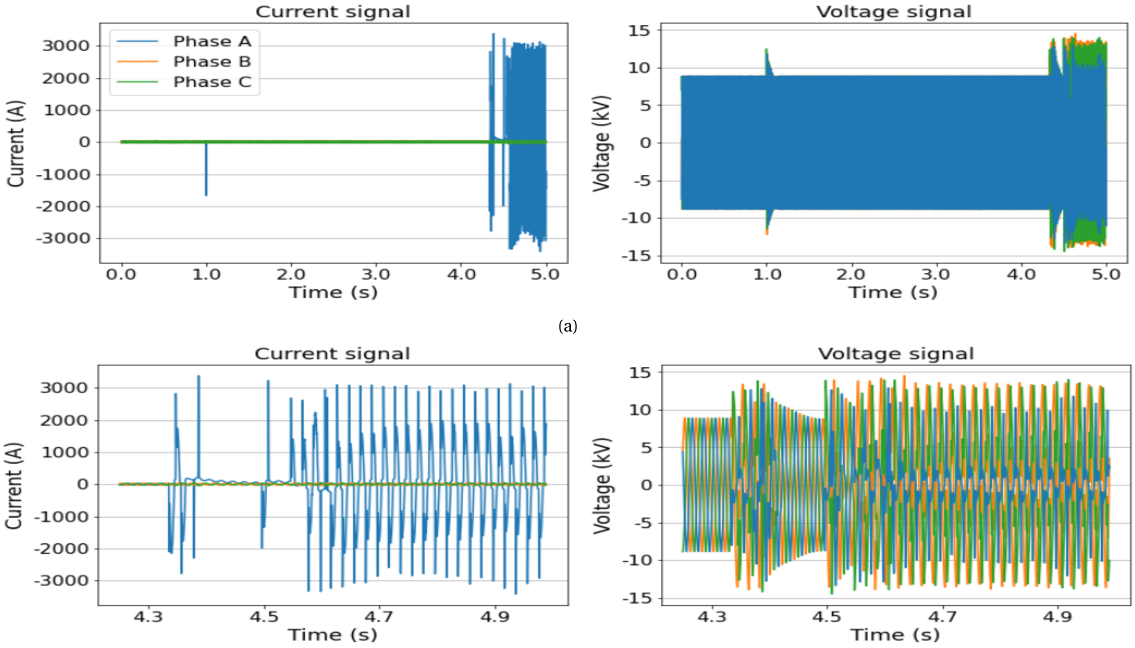
Compared to the fault classification model developed by Ranganathan in [2], the model has quite similar performance on the random unseen data it was tested regarding the weighted f1-score (95%) which can at least show that both models are equally good on that. As mentioned before the model developed in this thesis has more explainable and interpretable features that makes it more trustworthy to use in practice by Alliander. Additionally as shown in the previous section where tuning the stability model, it can certainly identify and locate accurately two-phase and three-phase faults whereas the model proposed by Ranganathan cannot.



(a)

(b)

Figure 5.11: Single-phase fault labeled as unstable by Alliander's algorithm and classified as single-phase fault by the proposed fault classification scheme (a) full signal (b) zoomed signal



(a)

(b)

Figure 5.12: Unstable fault that is still classified as unstable by the proposed fault classification scheme (a) full signal (b) zoomed signal

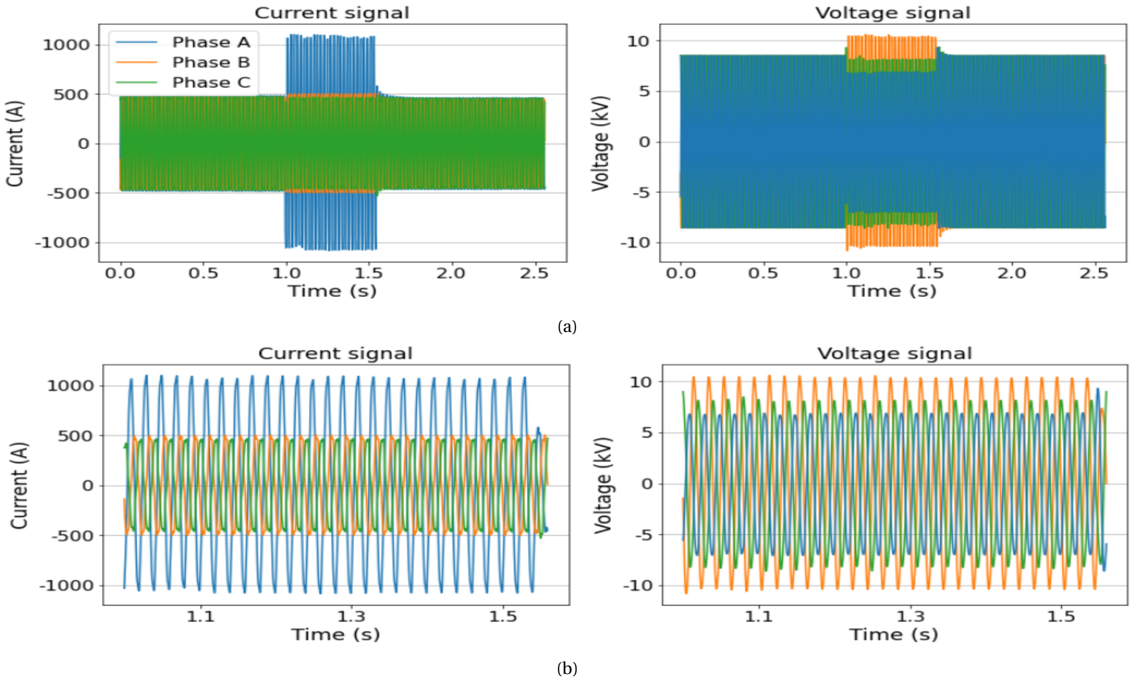


Figure 5.13: Single-phase fault labeled as unknown by Alliander's algorithm and classified as single-phase fault by the proposed fault classification scheme (a) full signal (b) zoomed signal

5.4. Summary

In this section, the development of the model that classifies and locates faults both stable and unstable ones was presented. In order to do that first a refinement of the stability thresholds was held which were evaluated based on the accuracy of the loop-impedance calculated in the end. The entire model is built consisted of the fault type classification, the stability check and the selection of part with the lowest loop-impedance. The results showed that the model is accurate enough and since can recognize better the unstable faults, it has the potential to replace the current Alliander fault location model. Finally, based on the analysis executed in this chapter, further research questions could be answered.

How can the unstable faults that can provide fault location be better distinguished from the ones that do not? In this chapter the development of new stability rules is done. The results showed that using these thresholds to evaluate the stability of a fault waveform, leads to a better distinction of faults that can provide a sufficiently accurate loop-impedance or -reactance (within the acceptable error margins) and hence a reliable fault location estimation. Otherwise these faults would be labeled as unstable faults and there would not be any possible fault location sent to the control room.

What are the methods to determine the best time period for the fault loop-impedance calculation for accurate fault location estimation? The model developed in this chapter includes also the calculation of the fault loop-impedance/-reactance. That means that the model can determine the best time period to calculate it. This is done after the fault type classification and the stability check. Similar to the stability check and to the same step at the current model of Alliander, the analysis is per period. For each segmented part, the impedance or reactance is calculated for the stable parts and then the period with the lowest impedance or reactance value based on the fault type is calculated. If a fault consists of one segmented section then this would be the final loop-impedance value, otherwise the suitable one of the two based on the rules that showed is selected.

6 Conclusion

The focus of this thesis was first to develop a machine learning (ML) model for the fault classification with explainable features that could recognize accurately the type of the fault in the distribution system. The fault type classification is an important step, before the estimation of the fault location and subsequently the restoration of the system. The fault location estimation is done based on the loop-impedance/-reactance comparison that the fault would give and from the simulation of the network. This loop-impedance/-reactance calculation is fault type dependent and thus a wrong type result would lead to a wrong fault location. So, it all comes to the ability to estimate the location of a fault quickly and accurately in order for the faulty section to get isolated and the power supply to get restored.

The explainability though is something that would not be considered a lot before, however in this thesis has great significance. In this thesis the ways of how a model can be explainable and interpretable are discovered and the suitable ones are selected. The artificial intelligence (AI) models are highly penetrated in all kinds of applications in industry. There are a lot of applications of critical importance that could not afford to turn into a black-box and not understand or even trust the result. So, the introduction of explainability in this thesis, shows how a ML fault classification model could be more interpretable so the operators trust the results of it.

Under these requirements different sets of explainable features were investigated so to form an appropriate and quite accurate classification model for the stable faults. Compared to a previous ML-based classification model, this model has similar accuracy and outperforms in terms of explainability. However, the classification model of Alliander works accurately also for the stable faults and it is explainable out of nature. The main issue comes from the unstable faults classification as the current rules seem too conservative. So, the thesis further focuses on incorporating the self-extinguishing and unstable faults recognition in the model, refining the stability rules.

Hence, a more general model is created which consists of a ML classification model and a per period stability check model. The model is extended into finding the final loop-impedance which would be compared with a simulation impedance, leading to a possible fault location. The evaluation of the stability thresholds developed is done based on the loop-impedance result compared to the corresponding ones given by Alliander. After that the whole model can be built and evaluated in a similar way. The results show that indeed the model developed can recognize efficiently the type of the faults, including self-extinguishing and unstable faults, and to classify correctly as stable faults previously labeled unstable faults. This is the main outcome and advantage of this model over the current Alliander one. Also, the performance of the model over the accurate loop-impedance/-reactance calculation are very promising.

6.1. Research Questions

The main research question for this thesis project is:

How can modern signal processing and machine learning techniques be used to construct an accurate and explainable fault classification scheme in the distribution network?

The sub-questions that have been answered during the course of this thesis and summarize the key findings are:

1. *Which explainability approaches exist for a machine learning model and how to evaluate them?*

   Based on the literature review held in chapter 2, the explainability can be categorized in three levels. The first is the pre-modeling explainability which mostly concerns the dataset and the feature engineering. Second, the explainability in the model that derives either out of its nature such as the Linear Model, Decision Tree (DT), kNN or a build-in hybrid model. Third, the post-modeling explainability which again can be distinguished in separate categories. It can be a model-agnostic or a model-specific explainability and global or local explainability. Model-agnostic is usually a black-box with specific formulation unrelated to the model that can give some explanations, whereas model-specific explainability is a model related that interferes in it. Global explainability is about the interpratation of the general response of the model whereas local refers to a single or a specific number of results. In the end the pre-modeling explainability is used for this project and in particular the feature engineering is focused only in explainable features. The choice of a model more explainable would come at expense of the accuracy since they are less powerful, thus this was not the approach followed. The post-modeling explainability was not selected because it is yet quite vague in its results, since different methods lead to different results.

   The evaluation of explainability/interpretability can be done by experts as an application level evaluation, by people non-experts in big scale but with less important tasks as a human level evaluation or by using an already certified explainable model as a proxy such as DT with a few levels in a function level evaluation. The choice of evaluation in this thesis was the application level evaluation based on experts which led accordingly the development of explainable features.

2. *What explainable features could be extracted for the fault classification?*

   In this thesis was very essential the development of explainable features. As seen by the literature review in chapter 2, in the majority of the classification models for faults in the transmission and the distribution system, the feature extraction is based on the Wavelet Transform (WT). However this affects the explainability of the model, frequency domain information (detailed coefficient) cannot provide precise frequency information. Hence that makes WT features less interpretable. So other more explainable techniques were explored and selected such as the Fourier Transform (FT), the sliding FT or the signal itself. Even though these provide poorer time-frequency resolution, within each window they can lead in a precise frequency content information. Using Short-time FT and more specifically the fundamental frequency component, that makes the per-period magnitude and phase of each waveform available. Subsequently that helps into shaping the symmetrical components, impedance, reactance and resistance, that could lead to interpretable features. These features were created using the average, minimum or maximum value of these components for the specific segmented part. Also in the case of the features developed based on the third harmonic magnitude's of current or voltage the energy is used. In chapter 4 the explainable feature sets developed can be seen.

3. *How can the unstable faults that can provide fault location be better distinguished from the ones that do not?*

   One of the most significant goals of this thesis project was to refine the stability rules for the unstable faults, since the existing ones are conservative. All the faults that can provide a reliable fault location estimation based on the loop-impedance, ideally should pass through the automatic procedure of finding this location. So, with the existed rules quite some faults of this kind are labeled as unstable and they cannot send a fault location to the control room. Locating unstable faults without any indication is quite more time-consuming taking into consideration the importance of time, when the power supply stops.

In chapter 5 new stability rules were developed. The results showed that using these thresholds more faults are classified as stable even though they were labeled as unstable based on Alliander current rules. This is an important finding of this thesis, since more of these unstable faults are sending a fault location estimation to the control room, thus the time needed for restoration of the network and the customer minutes lost would be shorter.

4. *What are the methods to evaluate the machine learning model's performance on classifying the faults?*

   This question concerns the performance of the fault type classification model that used Support Vector Machine (SVM). In chapter 4 the model works only for the stable faults where only one segmented part exists and the metric used for its evaluation is the macro f1-score which is a trade-off between precision and recall. Due to the use of a nested cross-validation, the evaluation is more objective since the model is validated in different sets in the end which it would not see in a straight split of training and validation set. So, the final result for the performance of each model comes from the average macro f1-score value on the different test set for all the five folds. Based on that all the models developed in chapter 4 had a quite good performance.

   Regarding the fault classification and location scheme developed in chapter 5, the use of ML is simply in the fault classification model. The ML model there is similar to the model with the symmetrical features in chapter 4, however the data used is different since the derived segmented parts could be even two. Again nested cross-validation is used and macro f1-score to evaluate and reach to the proper model selection, there is though a hold out of a test set that is used for the fault location evaluation. The performance of the ML model in classifying fault types is quite accurate and that is very promising since that influences the fault loop-impedance/-reactance calculation afterwards.

5. *What are the methods to determine the best time period for the fault-loop impedance calculation for accurate fault location estimation?*

   This question refers to the fault location which is the end goal of such model. In order to make that possible for all the kind of faults, the ML classification model alone is not sufficient, as the ones in chapter 4. That's why the loop-impedance correct calculation depends on the stability rules and the fault type. But the stability rules are also fault type dependent. So, an appropriate model is developed in chapter 5. It includes the fault type classification, the stability check and the selection of a suitable fault loop-impedance. The stability check and the fault loop-impedance calculation are done in a per-period analysis as in the current model of Alliander. For each segmented section and for their particular stable periods, the impedance and reactance are calculated for single-phase and multi-phase faults, respectively. The method to get the best time period for the fault loop-impedance is to select the period with the lowest impedance or reactance value correspondingly. Then if a fault consists of one segmented section then this would be the final loop-impedance value, otherwise the suitable one of the two based on the rules that showed is selected. The final calculated values were compared with the corresponding values from Alliander's current algorithm. Using the metrics root mean squred error (RMSE), mean absolute error (MAE) and 90th percentile the results were evaluated based on the error limits given. The results show that this model offers sufficiently accurate performance in ending up to the fault loop-impedance.

6.2. Future Research

The models developed and subsequently the outcome reached by this thesis was based on a number of limitations presented in separate areas throughout the study. In this section, these limitations are acknowledged and possible avenues for future research are presented.

Opposite to what most of the reviewed literature does, real-world fault data recordings from a distribution system are used for developing the models here. This inevitably corresponds to a limited number of reliable real fault data to explore and build the classification ML and the stability model. The existence of a larger sample space of faults available to investigate, would ensure broader range of real data for the models to trained and validated on which leads to the capture of greater variety of characteristics prevailed in the fault waveforms. Additionally, the fact that in the reviewed literature the majority of the papers are dealing with synthetic data that are quite purer than the real-world ones, creates a gap between the techniques used there and the techniques needed by a real application. So, the proposed methods based on synthetic data need to be investigated on how applicable are if the input is the real data. First, Alliander database is getting larger as more faults are stored, so it would be desirable in the future to retrain the models with the new extended dataset. Second, the exploitation of the powerful tool called Real Time Digital Simulator (RTDS) for creating synthetic data would be very helpful into extending the dataset suitably. With the RTDS the data created would have the closest to the real faults form achieved and thus the information providing would be very useful for training and validating the models.

Since the explainability and the interpretability of the results are essential for the development of the classification models in this project, the complexity and hence the accuracy of the model selected was limited. The ML technique selected here was the Support Vector Machine (SVM) which provides a sufficient a trade-off between explainability and classifying performance. As explainability and interepretability of AI and ML models is an emerging topic, if a suitable technique of post-modeling explainability arise that would let a more complex model to be picked up, such as a deep learning (DL) based one, that would possibly be able to exploit better the information gained from the dataset of faults given. Therefore, techniques as Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN) that can better spot and handle the physical characteristics of waveforms as an image and a time-series respectively, would be worth exploring.

Regarding the development of the new stability rules again the existence of a larger and reliable number of faults labeled as either only initially or fully unstable would let the model to be trained, validated and eventually tested in more fault data recordings. Moreover, most of the unstable fault entries existing concern single-phase faults since they occur in higher frequency than the multi-phase faults. That leads into developing a stability check model mainly based on the unstable single-phase faults, ending up in the same rules for the zero sequence and the phase-phase signals for the single-phase and multi-phase samples respectively. The availability of a greater amount of multi-phase faults would make possible the exploration of a separate set of rules, that could may be even result in more real stable multi-phase faults to be recognized correctly. So, again creating appropriate synthetic data with the help of RTDS would help into having a better trained model. Furthermore, the development of an optimization process to find the exact combination of the values for each stability feature would be more accurate. An alternative approach could be explored based on unsupervised learning techniques with a solid framework. This could focus on all the per-period stability characteristics of the fault data which is infeasible to manually develop labels for.

As mentioned before, self-extinguishing faults are faults that after some periods of fault existing, extinguish by themselves without any circuit breaker (CB) closing. That means that no immediate action is needed, however its location is good to be known so preventive measures and maintenance applied there and hence not a stable fault is developed later. In the full model developed in chapter 5 this class of faults is included in the training and testing stages and it seems that the model there works sufficiently into distinguishing self-extinguishing faults. Nevertheless in the rare occasion of such a fault with a developed stable part before, the model does not work as it should because it recognizes this part as a stable fault and the section rules are developed in a way to promote it over the self-extinguishing label. The existence of stable part that the model recognizes, it leads to a fault location that would be the correct one. Under that thought, perhaps a separate model for the distinction between the permanent faults and the self-extinguishing faults that after that would also investigate the possible existence of stable part. That way, this kind of self-extinguishing faults would not classified incorrectly as permanent ones.

In this thesis, the performance metrics used for the classification model was the multi-label f1-score and for the fault loop-impedance/-reactance the RMSE, MAE and 90th percentile. Because f1-score serves as a trade-off of precision and recall metrics, f1-score was chosen as suitable for evaluating with the same weight the type of misclassification (false positive or false negative). For the fault loop-impedance/-reactance results the combination of three metrics was based on having a good reflection on the average value of errors in different ways, with these specific metrics that seem from the literature to be appropriate. For both the cases, there is room to investigate more metrics that could be more suitable to evaluate the corresponding model.

Finally, this thesis was focused on what extent the signal processing and machine learning techniques would help into classifying and locating accurately faults in the distribution network. The computation time minimization was not under the scope of this project, however since it is considered for practical implementation, it could be useful to explore any improvement in terms of run time. The optimization of the code is one aspect that could be explored and the segmentation technique, a data preprocessing method that needs time. In particular, the segmentation algorithm could be further investigated by either building a different cost function, either by replacing it with a totally different technique that is not based on the signal deviation, either by replacing suitably the model so the segmentation could be discarded.

Bibliography

[1] Frans Provoost. “Intelligent distribution network design”. In: *Eindhoven: Eindhoven University of Technology* (2009). DOI: 10.6100/IR651978.

[2] Archana Ranganathan. “Automatic Identification of Fault Types in the Distribution Network using Supervised Learning”. In: (2021). URL: http://resolver.tudelft.nl/uuid:e8660f80-1aea-4fdb-929e-29c24a62a965.

[3] Tongyou Gu and Frans Provoost. “Improved fault location algorithm for MV networks based on practical experience”. In: *CIRED-Open Access Proceedings Journal* 2017.1 (2017), pp. 1211–1214. DOI: 10.1049/oap-cired.2017.1153.

[4] Tongyou Gu and Frans Provoost. “Improving network performance by recognition and location of self-extinguishing faults”. In: (2013). DOI: 10.1049/cp.2013.0728.

[5] Hans Wolse et al. “Experience and tendencies after 40 years outage data registration in the Netherlands”. In: *CIRED-Open Access Proc. J* 2017 (2017), pp. 2279–2282. DOI: 10.1049/oap-cired.2017.0251.

[6] Sophi Shilpa Gururajapathy, Hazlie Mokhlis, and Hazlee Azil Illias. “Fault location and detection techniques in power distribution systems with distributed generation: A review”. In: *Renewable and sustainable energy reviews* 74 (2017), pp. 949–958. DOI: 10.1016/j.rser.2017.03.021.

[7] Hamid Mirshekali et al. “Real fault location in a distribution network using smart feeder meter data”. In: *Energies* 14.11 (2021), p. 3242. DOI: 10.3390/en14113242.

[8] F Koehler, S Cobben, and F Provoost. “Self-extinguishing faults in mv cable networks-feasibility study of fault prediction”. In: *International Conference on Renewable Energies and Power Quality (ICREPQ’12)*. 2012. DOI: 10.24084/repqj10.210.

[9] Frans Provoost and Walter Van Buijtenen. “Practical experience with fault location in MV cable network”. In: *CIRED 2009-20th International Conference and Exhibition on Electricity Distribution-Part 1*. IET. 2009, pp. 1–4. DOI: 10.1049/cp.2009.0817.

[10] Christoph Molnar. *Interpretable machine learning*. Lulu. com, 2020. URL: https://christophm.github.io/interpretable-ml-book/.

[11] Isel del Carmen Grau García. “Self-labeling Grey-box Model: An Interpretable Semi-supervised Classifier”. PhD thesis. Vrije Universiteit Brussel, 2020. URL: https://ai.vub.ac.be/wp-content/uploads/2020/12/PhD_book.pdf.

[12] Bahador Khaleghi. “The How of Explainable AI: Pre-modelling Explainability”. In: *Towards Data Science* (2019). URL: https://towardsdatascience.com/the-how-of-explainable-ai-pre-modelling-explainability-699150495fe4.

[13] W James Murdoch et al. “Interpretable machine learning: definitions, methods, and applications”. In: *arXiv preprint arXiv:1901.04592* (2019). DOI: doi.org/10.48550/arXiv.1901.04592.

[14] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. “Model-agnostic interpretability of machine learning”. In: *arXiv preprint arXiv:1606.05386* (2016). DOI: 10.48550/arxiv.1606.05386.

[15] Scott M Lundberg and Su-In Lee. “A unified approach to interpreting model predictions”. In: *Advances in neural information processing systems* 30 (2017). URL: https://proceedings.neurips.cc/paper/2017/file/8a20a8621978632d76c43dfd28b67767-Paper.pdf.

[16] Finale Doshi-Velez and Been Kim. “Towards a rigorous science of interpretable machine learning”. In: *arXiv preprint arXiv:1702.08608* (2017). DOI: 10.48550/ARXIV.1702.08608.

[17] Satyapriya Krishna et al. “The Disagreement Problem in Explainable Machine Learning: A Practitioner’s Perspective”. In: *arXiv preprint arXiv:2202.01602* (2022). DOI: 10.48550/arxiv.2202.01602.

[18] Xolani G Magagula et al. “A fault classification and localization method in a power distribution network”. In: *2017 IEEE AFRICON*. IEEE. 2017, pp. 1337–1343. DOI: 10.1109/AFRCON.2017.8095676.

[19] Hanif Livani and Cansın Yaman Evrenosoğlu. “A fault classification method in power systems using DWT and SVM classifier”. In: *Pes T&d 2012*. IEEE. 2012, pp. 1–5. DOI: 10.1109/TDC.2012.6281686.

[20] Hanif Livani and Cansın Yaman Evrenosoğlu. “A fault classification and localization method for three-terminal circuits using machine learning”. In: *IEEE Transactions on Power Delivery* 28.4 (2013), pp. 2282–2290. DOI: 10.1109/TPWRD.2013.2272936.

[21] Tamer S Abdelgayed, Walid G Morsi, and Tarlochan S Sidhu. “Fault detection and classification based on co-training of semisupervised machine learning”. In: *IEEE Transactions on Industrial Electronics* 65.2 (2017), pp. 1595–1605. DOI: 10.1109/TIE.2017.2726961.

[22] Kapildev Lout and Raj K Aggarwal. “A feedforward Artificial Neural Network approach to fault classification and location on a 132kV transmission line using current signals only”. In: *2012 47th International Universities Power Engineering Conference (UPEC)*. IEEE. 2012, pp. 1–6. DOI: 10.1109/UPEC.2012.6398574.

[23] Yılmaz Aslan and Yunus Emre Yağan. “Artificial neural-network-based fault location for power distribution lines using the frequency spectra of fault data”. In: *Electrical Engineering* 99.1 (2017), pp. 301–311. DOI: 10.1007/s00202-016-0428-8.

[24] Jun Han et al. “Fault type classification in transmission line using STFT”. In: (2012). DOI: 10.1049/cp.2012.0130.

[25] Moez Ben Hessine, Houda Jouini, and Souad Chebbi. “Fault detection and classification approaches in transmission lines using artificial neural networks”. In: *MELECON 2014-2014 17th IEEE Mediterranean Electrotechnical Conference*. IEEE. 2014, pp. 515–519. DOI: 10.1109/MELCON.2014.6820588.

[26] K Gayathri and N Kumarappan. “Comparative Study of Fault Identification and Classification on EHV Lines Using Discrete Wavelet Transform and Fourier Transform Based”. In: (2008). DOI: 10.5281/zenodo.1329252.

[27] D Thukaram, HP Khincha, and HP Vijaynarasimha. “Artificial neural network and support vector machine approach for locating faults in radial distribution systems”. In: *IEEE transactions on power delivery* 20.2 (2005), pp. 710–721. DOI: 10.1109/TPWRD.2005.844307.

[28] Yong Sheng and Steven M Rovnyak. “Decision tree-based methodology for high impedance fault detection”. In: *IEEE Transactions on Power Delivery* 19.2 (2004), pp. 533–536. DOI: 10.1109/TPWRD.2003.820418.

[29] RN Mahanty and PB Dutta Gupta. “Comparison of fault classification methods based on wavelet analysis and ANN”. In: *Electric Power Components and Systems* 34.1 (2006), pp. 47–60. DOI: 10.1080/15325000691001485.

[30] Šejla Džakmić, Tarik Namas, and Alma Husagić-Selman. “Combined Fourier transform and Mexican hat wavelet for fault detection in distribution networks”. In: *2017 9th IEEE-GCC Conference and Exhibition (GCCCE)*. IEEE. 2017, pp. 1–6. DOI: 10.1109/IEEEGCC.2017.8447905.

[31] Moez Ben Hessine, Sahbi Marrouchi, and Souad Chebbi. “A fault classification scheme with high robustness for transmission lines using fuzzy-logic system”. In: *2017 International Conference on Advanced Systems and Electric Technologies (IC_ASET)*. IEEE. 2017, pp. 256–261. DOI: 10.1109/ASET.2017.7983701.

[32] Papia Ray and Debani Prasad Mishra. “Support vector machine based fault classification and location of a long transmission line”. In: *Engineering science and technology, an international journal* 19.3 (2016), pp. 1368–1380. DOI: 10.1016/j.jestch.2016.04.001.

[33] N Ramesh Babu and B Jagan Mohan. “Fault classification in power systems using EMD and SVM”. In: *Ain Shams Engineering Journal* 8.2 (2017), pp. 103–111. DOI: 10.1016/j.asej.2015.08.005.

[34] R Subhashree, C Sarada Preethi, and P Supriya. “Fault distance identification in transmission line using STFT algorithm”. In: *2016 International Conference on Computer Communication and Informatics (ICCCI)*. IEEE. 2016, pp. 1–4. DOI: 10.1109/ICCCI.2016.7480036.

[35] Yew Ming Yeap and Abhisek Ukil. “Fault detection in HVDC system using short time Fourier transform”. In: *2016 IEEE Power and Energy Society General Meeting (PESGM)*. IEEE. 2016, pp. 1–5. DOI: 10.1109/PESGM.2016.7741323.

[36] Debrup Das, Neeraj Kumar Singh, and Avinash K Sinha. “A comparison of Fourier transform and wavelet transform methods for detection and classification of faults on transmission lines”. In: *2006 IEEE Power India Conference*. IEEE. 2006, 7–pp. DOI: 10.1109/POWERI.2006.1632580.

[37] Vladimir Nasteski. “An overview of the supervised machine learning methods”. In: *HORIZONS.B* 4 (Dec. 2017), pp. 51–62. DOI: 10.20544/HORIZONS.B.04.1.17.P05.

[38] Mohamed Alloghani et al. “A systematic review on supervised and unsupervised machine learning algorithms for data science”. In: *Supervised and unsupervised learning for data science* (2020), pp. 3–21. DOI: 10.1007/978-3-030-22475-2_1.

[39] Alaa Tharwat. “Parameter investigation of support vector machine classifier with kernel functions”. In: *Knowledge and Information Systems* 61.3 (2019), pp. 1269–1302. DOI: 10.1007/s10115-019-01335-4.

[40] Ajay Mathur and Giles M Foody. “Multiclass and binary SVM classification: Implications for training and classification users”. In: *IEEE Geoscience and remote sensing letters* 5.2 (2008), pp. 241–245. DOI: 10.1109/LGRS.2008.915597.

[41] F. Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830. DOI: 10.48550/ARXIV.1201.0490.

[42] Peter Bruce, Andrew Bruce, and Peter Gedeck. *Practical statistics for data scientists: 50+ essential concepts using R and Python*. O’Reilly Media, 2020. URL: https://www.oreilly.com/library/view/practical-statistics-for/9781492072935/.

[43] Ajay Kulkarni, Deri Chong, and Feras A. Batarseh. “5 - Foundations of data imbalance and solutions for a data democracy”. In: *Data Democracy*. Academic Press, 2020, pp. 83–106. DOI: 10.1016/B978-0-12-818366-3.00005-8.

[44] Margherita Grandini, Enrico Bagli, and Giorgio Visani. “Metrics for multi-class classification: an overview”. In: *arXiv preprint arXiv:2008.05756* (2020). DOI: 10.48550/arxiv.2008.05756.

[45] Charles Truong, Laurent Oudre, and Nicolas Vayatis. “Selective review of offline change point detection methods”. In: *Signal Processing* 167 (2020), p. 107299. DOI: 10.1016/j.sigpro.2019.107299.

[46] Piotr Fryzlewicz. “Wild binary segmentation for multiple change-point detection”. In: *The Annals of Statistics* 42.6 (2014), pp. 2243–2281. DOI: 10.1214/14-AOS1245.

[47] Doron L Bergman. “Symmetry constrained machine learning”. In: *Proceedings of SAI Intelligent Systems Conference*. Springer. 2019, pp. 501–512. DOI: 10.1007/978-3-030-29513-4_37.

[48] Mahesh Pal and Giles M Foody. “Feature selection for classification of hyperspectral data by SVM”. In: *IEEE Transactions on Geoscience and Remote Sensing* 48.5 (2010), pp. 2297–2307. DOI: 10.1109/TGRS.2009.2039484.

[49] Jiliang Tang, Salem Alelyani, and Huan Liu. “Feature selection for classification: A review”. In: *Data classification: Algorithms and applications* (2014), p. 37. DOI: 10.1201/b17320.

[50] Jason Brownlee. “How to Choose a Feature Selection Method For Machine Learning”. In: *Machine Learning Mastery* (2019). URL: https://machinelearningmastery.com/feature-selection-with-real-and-categorical-data/.

[51] Max Kuhn, Kjell Johnson, et al. *Applied predictive modeling*. Vol. 26. Springer, 2013. DOI: 10.1007/978-1-4614-6849-3.

[52] Jean Golay, Michael Leuenberger, and Mikhail Kanevski. “Feature selection for regression problems based on the Morisita estimator of intrinsic dimension”. In: *Pattern Recognition* 70 (2017), pp. 126–138. DOI: 10.1016/j.patcog.2017.05.008.

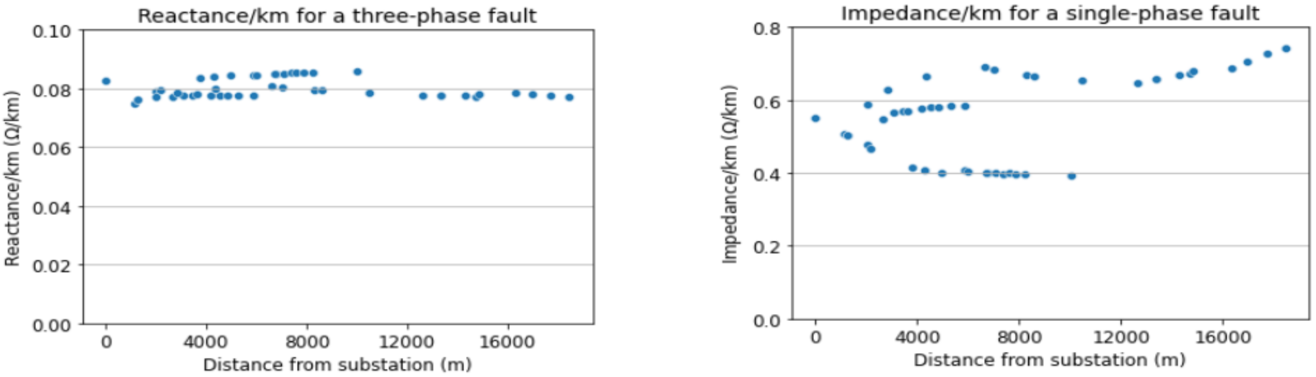
[53] Tae Kyun Kim. “Understanding one-way ANOVA using conceptual figures”. In: *Korean journal of anesthesiology* 70.1 (2017), pp. 22–26. DOI: 10.4097/kjae.2017.70.1.22.

[54] M Cover Thomas and A Thomas Joy. “Entropy, Relative Entropy, and Mutual Information”. In: *Elements of information theory*. John Wiley Sons, Ltd, 2006. Chap. 2, pp. 13–55. DOI: https://doi.org/10.1002/047174882X.ch2.

[55] Gareth James et al. *An introduction to statistical learning*. Vol. 112. Springer, 2013. DOI: 10.1007/978-1-4614-7138-7.

A

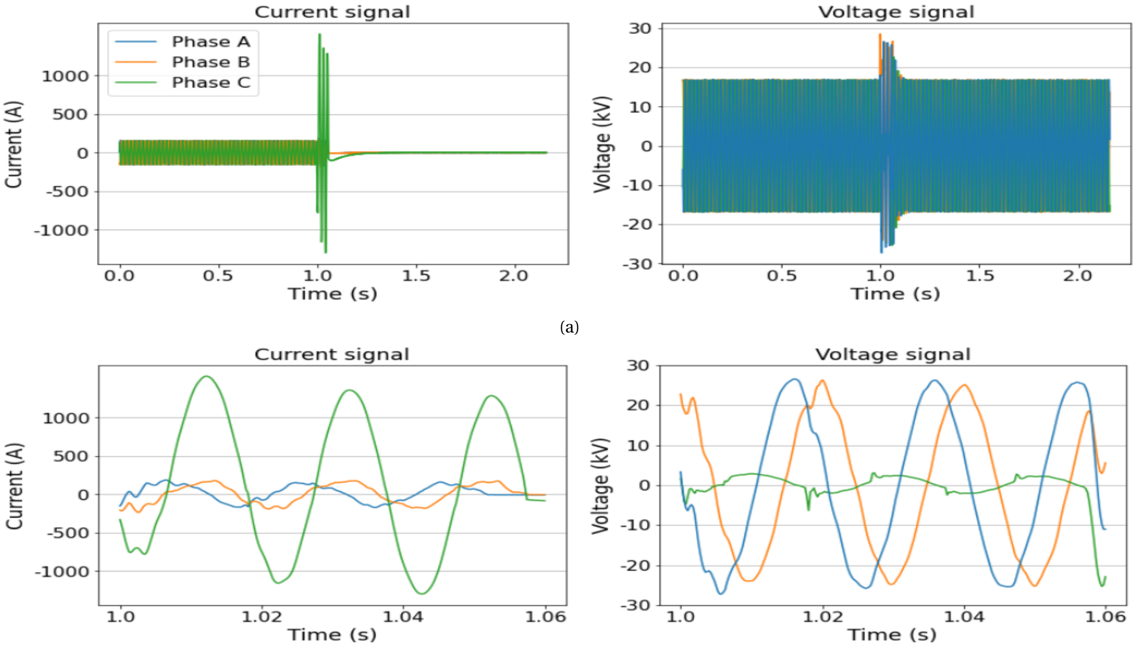
Supplementary Figures



(a) Scatter plot of the reactance/km for a three-phase fault

(b) Scatter plot of the impedance/km for a single-phase fault

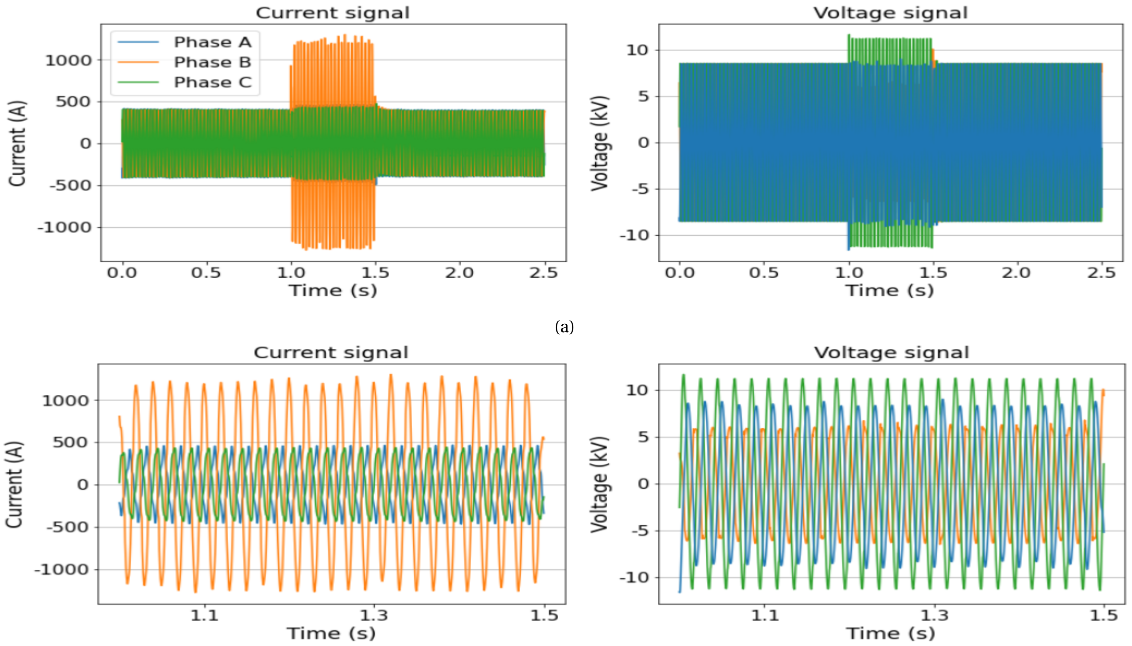
Figure A.1: Reactance/km and impedance/km for three- and single-phase fault respectively per distance of a fault from the substation. This concerns an example based on real observations of a specific substation. The greater the distance the further from the substation is the location, which means that the cables are become smaller in diameter, thus they have higher resistance than before. As it can be observed, the reactance/km has quite a constant value at around $0.08\Omega/km$ that remain unaffected by the distance. In contrast, the impedance/km deviates more between the values $0.4-0.8\Omega/km$ and at the furthest point rises to the highest values, as the cable has lower diameter



(a)

(b)

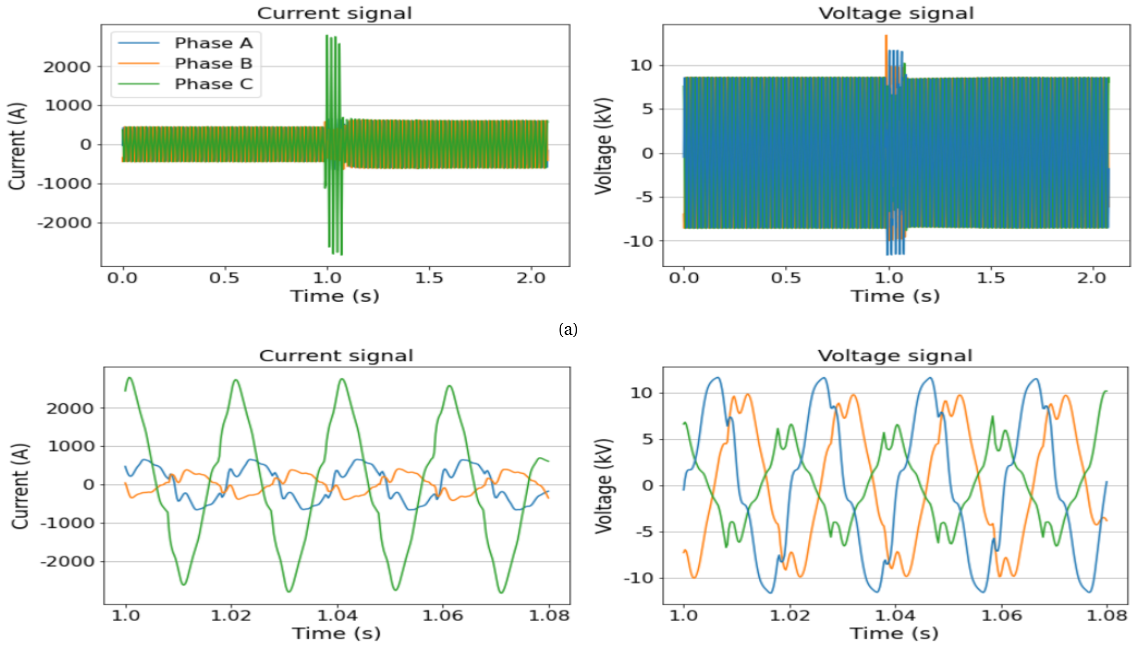
Figure A.2: Single-phase fault labeled as unstable by Alliander's algorithm and classified as single-phase fault by the proposed fault classification scheme (a) full signal (b) zoomed signal



(a)

(b)

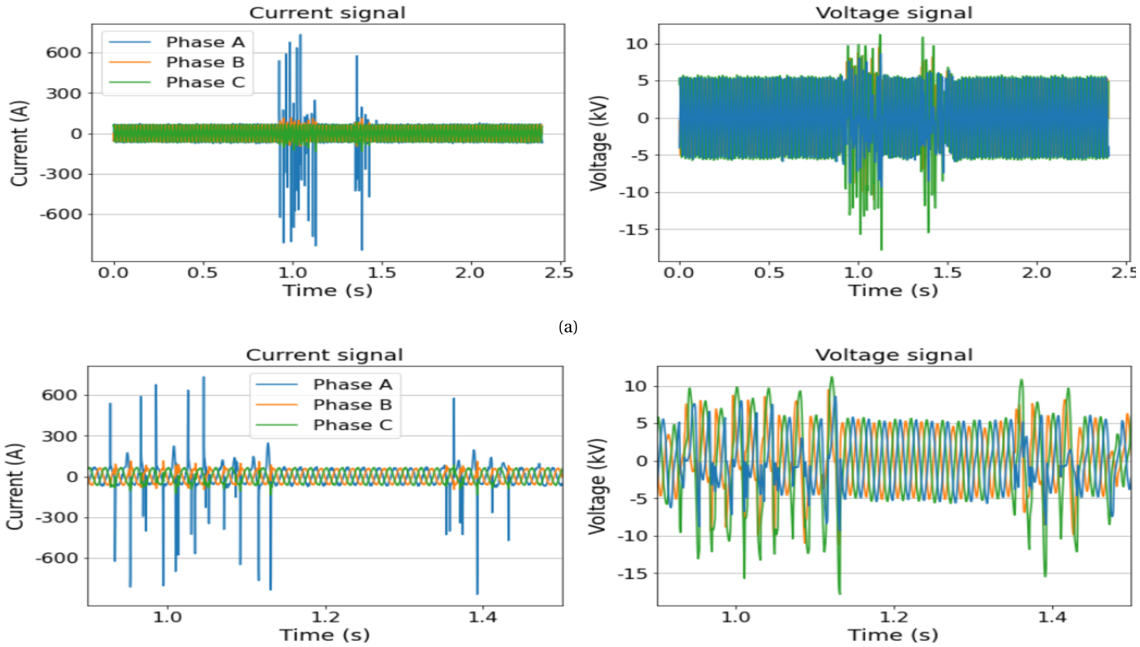
Figure A.3: Single-phase fault labeled as unstable by Alliander's algorithm and classified as single-phase fault by the proposed fault classification scheme (a) full signal (b) zoomed signal



(a)

(b)

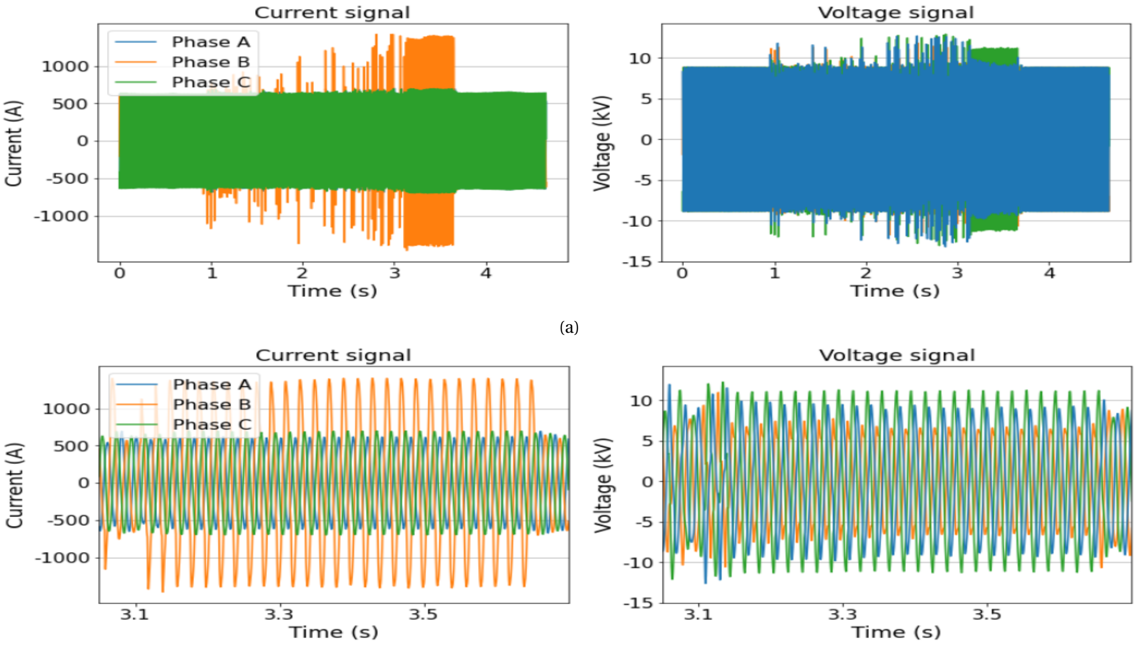
Figure A.4: Single-phase fault labeled as unstable by Alliander's algorithm and classified as single-phase fault by the proposed fault classification scheme (a) full signal (b) zoomed signal



(a)

(b)

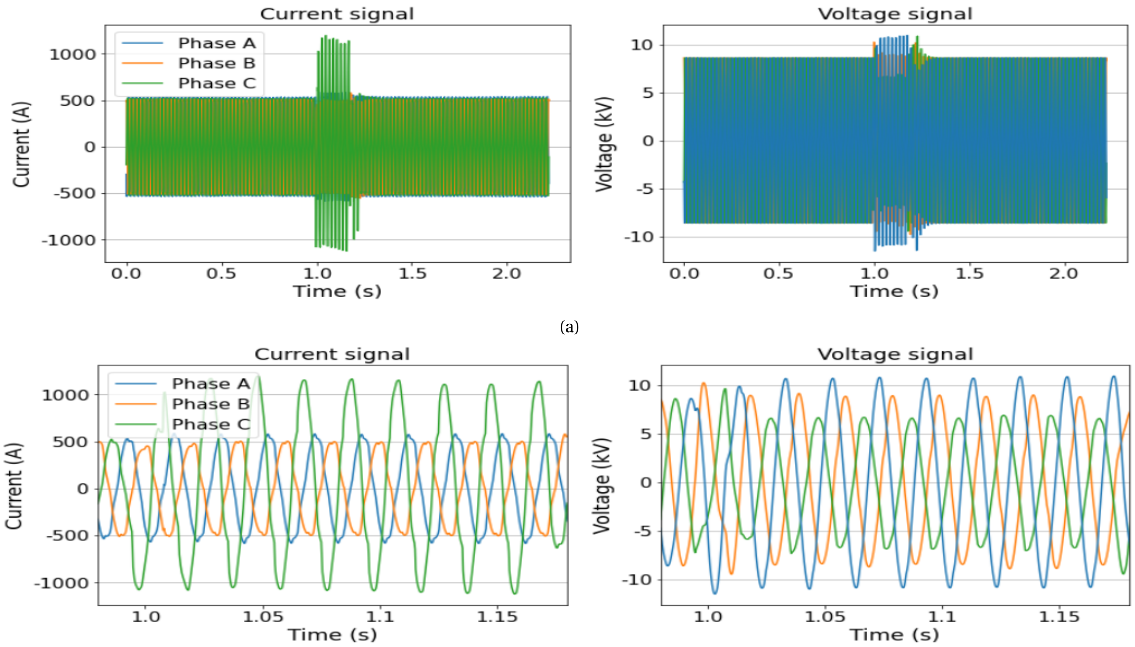
Figure A.5: Unstable fault labeled as unstable by Alliander's algorithm and classified as unstable fault by the proposed fault classification scheme (a) full signal (b) zoomed signal



(a)

(b)

Figure A.6: Single-phase fault labeled as unknown by Alliander's algorithm and classified as single-phase fault by the proposed fault classification scheme (a) full signal (b) zoomed signal



(a)

(b)

Figure A.7: Single-phase fault labeled as unknown by Alliander's algorithm and classified as single-phase fault by the proposed fault classification scheme (a) full signal (b) zoomed signal

B

Supplementary Results

Table B.1: Labels and impedance values for each data sample belonging to the final test set

DFR file name	True fault type	Current algorithm's label	Proposed algorithm's label	Current algorithm's impedance	Proposed algorithm's impedance
AMV_010V145_20190502_144308.dfr	single-ph	single-ph	single-ph	(3.932+1.809j)	(3.932+1.809j)
EDM_133_20190320_161428.dfr	single-ph	single-ph	single-ph	(14.1+8.316j)	(14.312+8.215j)
EDM_137_20180110_043646.dfr	single-ph	single-ph	single-ph	(4.859+2.703j)	(4.952+2.775j)
GRD_11_20190726_231807.dfr	single-ph	single-ph	single-ph	(20.121+10.652j)	(20.121+10.652j)
HAS1_2.10_20200525_1304067.dfr	single-ph	single-ph	single-ph	(10.104+5.394j)	(9.19+5.571j)
HHWN_106_20190607_205722.dfr	single-ph	single-ph	single-ph	(2.467+1.327j)	(2.467+1.327j)
HMM_156_20200817_134211.dfr	single-ph	single-ph	single-ph	(7.651+4.975j)	(7.069+4.817j)
HRNH_138_20180528_165157.dfr	single-ph	single-ph	single-ph	(29.698+10.929j)	(29.252+10.971j)
IJM_156_20191220_124400.dfr	single-ph	single-ph	single-ph	(31.891+12.819j)	(31.891+12.819j)
LEIN_V104_20190728_085715.dfr	single-ph	single-ph	two-ph	(11.732+4.431j)	(1.13+0.553j)
LEIN_V109_20180727_222558.dfr	single-ph	single-ph	single-ph	(5.977+2.718j)	(5.977+2.718j)
MNZL_1_20191227_174137.dfr	single-ph	single-ph	single-ph	(4.942+1.94j)	(4.942+1.94j)
NK1_2.14_20181113_152647.dfr	single-ph	single-ph	single-ph	(1.279+0.422j)	(1.279+0.422j)
OHK_1.18_20180220_034301.dfr	single-ph	single-ph	single-ph	(42.253+12.494j)	(41.896+12.305j)
OWD_19_20180614_235805.dfr	single-ph	single-ph	single-ph	(6.024+3.501j)	(6.012+3.473j)
OWD_19_20180615_010031.dfr	single-ph	single-ph	single-ph	(5.888+3.381j)	(5.888+3.381j)
RDW_010V203_20180710_021313.dfr	single-ph	single-ph	single-ph	(13.256+7.302j)	(12.75+7.595j)
RDW_010V203_20210125_225629.dfr	single-ph	single-ph	single-ph	(6.73+3.693j)	(6.73+3.693j)
RW_010V116_20190913_124456.dfr	single-ph	single-ph	single-ph	(24.769+12.274j)	(24.769+12.274j)
SCH1_138_20190418_093922.dfr	single-ph	single-ph	single-ph	(1.773+0.879j)	(1.773+0.879j)
SCH1_177_20200809_164556.dfr	single-ph	single-ph	single-ph	(5.76+3.071j)	(5.76+3.071j)
SK_3_20180108_140918.dfr	single-ph	single-ph	single-ph	(5.153+4.227j)	(5.153+4.227j)
SK_3_20180221_141257.dfr	single-ph	single-ph	single-ph	(3.189+1.696j)	(3.192+1.603j)
SK_3_20180814_202229.dfr	single-ph	single-ph	single-ph	(6.108+2.992j)	(5.907+2.961j)
SK_3_20181024_104728.dfr	single-ph	single-ph	single-ph	(3.189+1.27j)	(3.189+1.27j)
SK_3_20190524_075328.dfr	single-ph	single-ph	single-ph	(28.803+12.856j)	(28.803+12.856j)
WHS_010V270_20180930_013556.dfr	single-ph	single-ph	single-ph	(2.505+3.18j)	(2.505+3.18j)
ALPV_V0102_20210218_122712.dfr	two-ph	two-ph	two-ph	(0.264+0.05j)	(0.264+0.05j)
AMV_010V133_20131014_223930.dfr	two-ph	two-ph	two-ph	(2.436+0.502j)	(2.376+0.508j)
BGM_1.13_20180503_145635.dfr	two-ph	two-ph	two-ph	(1.832+0.986j)	(1.832+0.986j)
BGM_2.18_20180203_032422.dfr	two-ph	two-ph	two-ph	(2.039+0.527j)	(2.064+0.526j)
DK_11_20180310_193352.dfr	two-ph	two-ph	two-ph	(1.43+0.455j)	(1.43+0.455j)
DK_22_20190603_113141.dfr	two-ph	two-ph	two-ph	(3.649+0.663j)	(3.685+0.655j)
EDM_010V201_20200626_101142.dfr	two-ph	two-ph	two-ph	(0.144+0.078j)	(0.15+0.078j)
EDM_137_20180420_024922.dfr	two-ph	two-ph	two-ph	(1.564+0.748j)	(1.56+0.745j)
EDM_144_20140715_090706.dfr	two-ph	two-ph	two-ph	(1.086+0.459j)	(1.074+0.448j)
HAS1_2.09_20180802_104815.dfr	two-ph	two-ph	two-ph	(0.531+0.666j)	(0.531+0.666j)
HMM_144_20190125_102140.dfr	two-ph	two-ph	two-ph	(0.5+0.3j)	(0.512+0.245j)
IJM_145_20200810_011134.dfr	two-ph	two-ph	two-ph	(0.561+0.156j)	(0.556+0.156j)
LEI_RU_2_010V269_20180531_131719.dfr	two-ph	two-ph	two-ph	(0.205+0.087j)	(0.205+0.087j)
LEI_ZW_124_20170616_220612.dfr	two-ph	two-ph	three-ph	(1.011+0.55j)	(1.01+0.546j)
LEI_ZW_128_20200812_081739.dfr	two-ph	two-ph	two-ph	(1.144+0.518j)	(1.144+0.518j)
LMR_2.17_20200721_171531.dfr	two-ph	two-ph	two-ph	(0.992+0.42j)	(0.992+0.42j)
NK1_2.10_20190324_032617.dfr	two-ph	two-ph	two-ph	(0.513+0.359j)	(0.513+0.359j)
NRD_147_20200821_102156.dfr	two-ph	two-ph	two-ph	(0.392+0.166j)	(0.419+0.143j)
OWD_14_20181002_100655.dfr	two-ph	two-ph	two-ph	(1.994+0.431j)	(1.994+0.431j)

TEX_010V145_20160201_042343.dfr	two-ph	two-ph	two-ph	(1.083+0.309j)	(1.036+0.301j)
WOM_3_20190805_021058.dfr	two-ph	two-ph	two-ph	(3.526+0.53j)	(3.526+0.53j)
WV_9_20180103_205324.dfr	two-ph	two-ph	two-ph	(0.341+0.147j)	(0.341+0.147j)
ZVH1_158_20150424_071500.dfr	two-ph	two-ph	two-ph	(0.971+0.734j)	(0.975+0.728j)
DTN_16_20190728_215010.dfr	three-ph	three-ph	three-ph	(0.037+0.016j)	(0.037+0.016j)
HAS1_2.11_20180717_105041.dfr	three-ph	three-ph	three-ph	(0.704+0.755j)	(0.736+0.697j)
HAS1_2.11_20190731_134839.dfr	three-ph	three-ph	three-ph	(0.747+0.747j)	(0.744+0.76j)
HMM_137_20190318_093228.dfr	three-ph	three-ph	three-ph	(0.8+0.488j)	(0.767+0.484j)
LEIN_V120_20180703_055926.dfr	three-ph	three-ph	three-ph	(0.325+0.167j)	(0.321+0.162j)
LEI_ZW_202_20180728_064056.dfr	three-ph	three-ph	three-ph	(0.701+0.324j)	(0.701+0.324j)
NRD_144_20190625_060913.dfr	three-ph	three-ph	three-ph	(0.85+0.387j)	(0.875+0.364j)
OHK_1.17_20201103_103619.dfr	three-ph	three-ph	three-ph	(2.872+1.081j)	(2.871+1.072j)
PSW_10_20200127_121533.dfr	three-ph	three-ph	three-ph	(0.606+0.315j)	(0.606+0.315j)
RW_010V116_20200627_011901.dfr	three-ph	three-ph	three-ph	(1.839+0.348j)	(1.839+0.348j)
SCH1_139_20180702_151226.dfr	three-ph	three-ph	three-ph	(1.992+0.903j)	(1.985+0.896j)
SCH1_139_20181128_205005.dfr	three-ph	three-ph	three-ph	(1.222+0.59j)	(1.222+0.59j)
SK_3_20200922_013409.dfr	three-ph	three-ph	three-ph	(0.248+0.131j)	(0.252+0.126j)
WOR_14_20200705_133003.dfr	three-ph	three-ph	three-ph	(0.661+0.106j)	(0.677+0.098j)
WSP_139_20201212_161344.dfr	three-ph	three-ph	three-ph	(0.229+0.072j)	(0.229+0.072j)
AH_2.04_20181029_161149.dfr	self-ext	self-ext	self-ext	(6.247+22.367j)	-
AH_2.13_20200419_172903.dfr	self-ext	self-ext	self-ext	(5.467+24.072j)	-
ALKB_106_20200604_084437.dfr	self-ext	self-ext	self-ext	(298.255+40.055j)	-
ALPC2_V0206_20210130_212530.dfr	self-ext	self-ext	self-ext	(280.648+84.431j)	-
ALPC2_V0206_20210130_212546.dfr	self-ext	self-ext	self-ext	(192.096+64.449j)	-
AMV_010V133_20180610_010507.dfr	self-ext	self-ext	self-ext	(24.527+32.577j)	-
BGM_1.06_20190501_124457.dfr	self-ext	self-ext	self-ext	(45.424+52.931j)	-
BGM_1.06_20210225_001534.dfr	self-ext	self-ext	self-ext	(20.623+28.528j)	-
BGM_1.06_20210225_001643.dfr	self-ext	self-ext	self-ext	(22.379+14.605j)	-
BGM_1.06_20210225_001650.dfr	self-ext	self-ext	self-ext	(26.269+39.736j)	-
DK20_020V112_20180225_083634.dfr	self-ext	self-ext	self-ext	(50+50j)	-
DK_22_20190208_040828.dfr	self-ext	self-ext	self-ext	(233.965+188.647j)	-
DTN_13_20190621_142431.dfr	self-ext	self-ext	self-ext	(11.629+27.7j)	-
DTN_18_20200312_081832.dfr	self-ext	self-ext	self-ext	(12.119+28.04j)	-
ELT_203_20180610_022144.dfr	self-ext	self-ext	self-ext	(55.816+79.921j)	-
ELT_205_20201117_114002.dfr	self-ext	self-ext	self-ext	(194.869+126.638j)	-
HBY_1.07_20180918_161840.dfr	self-ext	self-ext	self-ext	(14.229+15.647j)	-
HMM_138_20191211_162710.dfr	self-ext	self-ext	self-ext	(461.466+145.349j)	-
HTM_2.02_20190219_164807.dfr	self-ext	self-ext	self-ext	(69.291+67.335j)	-
HTM_2.02_20190220_192658.dfr	self-ext	self-ext	self-ext	(17.525+37.247j)	-
HTM_2.02_20190221_051950.dfr	self-ext	self-ext	self-ext	(18.911+40.664j)	-
HTM_2.02_20190222_000124.dfr	self-ext	self-ext	self-ext	(19.358+43.335j)	-
HTM_2.02_20190223_105907.dfr	self-ext	self-ext	self-ext	(20.315+43.679j)	-
HTM_2.02_20190223_163956.dfr	self-ext	self-ext	self-ext	(14.057+35.518j)	-
HTM_2.02_20190224_231640.dfr	self-ext	self-ext	self-ext	(14.375+37.439j)	-
HTM_2.02_20190307_011511.dfr	self-ext	self-ext	self-ext	(19.831+29.324j)	-
HTM_2.02_20190307_094507.dfr	self-ext	self-ext	self-ext	(69.428+72.297j)	-
HTM_2.13_20190215_001501.dfr	self-ext	self-ext	self-ext	(73.444+49.598j)	-
HTM_2.13_20190302_054932.dfr	self-ext	self-ext	self-ext	(39.782+55.461j)	-
HTM_2.13_20190302_142927.dfr	self-ext	self-ext	self-ext	(60.813+50.83j)	-
HTM_2.19_20200119_100055.dfr	self-ext	self-ext	self-ext	(140.927+80.125j)	-
HTM_2.19_20200128_001253.dfr	self-ext	self-ext	self-ext	(50+50j)	-
HTM_2.25_20201205_205228.dfr	self-ext	self-ext	self-ext	(104.349+110.673j)	-
LEIN_V109_20190726_190420.dfr	self-ext	self-ext	self-ext	(18.218+16.3j)	-

LEI_vzw_2_20200429_223951.dfr	self-ext	self-ext	unstable	(50+50j)	-
LMR_1.07_20180729_024729.dfr	self-ext	self-ext	self-ext	(136.87+104.071j)	-
LMR_1.07_20180805_043150.dfr	self-ext	self-ext	self-ext	(131.005+117.219j)	-
LMR_1.07_20180805_043321.dfr	self-ext	self-ext	self-ext	(186.218+167.222j)	-
LMR_1.07_20180805_043415.dfr	self-ext	self-ext	self-ext	(219.366+145.075j)	-
LMR_1.09_20180621_162131.dfr	self-ext	self-ext	self-ext	(92.004+99.387j)	-
LMR_1.09_20210822_142612.dfr	self-ext	self-ext	unstable	(8.701+5.439j)	-
LMR_1.09_20210822_142619.dfr	self-ext	self-ext	unstable	(7.489+4.734j)	-
LMR_2.07_20200625_171053.dfr	self-ext	self-ext	self-ext	(21.285+15.499j)	-
LMR_2.08_20200602_122818.dfr	self-ext	self-ext	self-ext	(25.523+26.109j)	-
LMR_2.17_20181111_053152.dfr	self-ext	self-ext	self-ext	(65.342+39.476j)	-
LMR_2.17_20181225_204004.dfr	self-ext	self-ext	self-ext	(88.201+110.163j)	-
MNZL1_010V109_20191015_153659.dfr	self-ext	self-ext	self-ext	(152.05+180.901j)	-
MNZL1_010V113_20190221_111641.dfr	self-ext	self-ext	self-ext	(447.473+54.035j)	-
MNZL1_010V113_20190317_000304.dfr	self-ext	self-ext	self-ext	(302.431+99.07j)	-
MNZL1_010V129_20201231_212604.dfr	self-ext	self-ext	self-ext	(75.024+61.705j)	-
NK1_2.18_20191004_071652.dfr	self-ext	self-ext	self-ext	(8.29+32.754j)	-
NK1_2.18_20191004_200827.dfr	self-ext	self-ext	self-ext	(16.744+61.242j)	-
NK1_2.18_20200316_110920.dfr	self-ext	self-ext	self-ext	(50+50j)	-
NK1_2.18_20201008_185847.dfr	self-ext	self-ext	self-ext	(12.957+55.881j)	-
NK1_2.18_20201008_185903.dfr	self-ext	self-ext	self-ext	(13.892+54.912j)	-
NK1_2.18_20201008_185913.dfr	self-ext	self-ext	self-ext	(21.483+82.454j)	-
NK1_2.18_20201101_021358.dfr	self-ext	self-ext	self-ext	(42.287+108.324j)	-
NK1_2.18_20201109_060902.dfr	self-ext	self-ext	self-ext	(561.114+223.086j)	-
NK1_2.18_20201109_061333.dfr	self-ext	self-ext	self-ext	(6.876+45.301j)	-
NK1_2.18_20201109_061829.dfr	self-ext	self-ext	self-ext	(16.585+217.656j)	-
OHK_1.13_20200817_005230.dfr	self-ext	self-ext	self-ext	(109.581+180.189j)	-
OWD_17_20191127_081433.dfr	self-ext	self-ext	self-ext	(85.885+88.187j)	-
PSW_18_20180308_084900.dfr	self-ext	self-ext	self-ext	(50+50j)	-
RDW_010V203_20210125_225624.dfr	self-ext	self-ext	self-ext	(71.385+70.235j)	-
RW_010V106_20191011_181810.dfr	self-ext	self-ext	self-ext	(200.72+104.581j)	-
SK_3_20180511_173257.dfr	self-ext	self-ext	self-ext	(26.213+44.361j)	-
TEX_010V137_20180205_212740.dfr	self-ext	self-ext	self-ext	(60.319+33.899j)	-
TEX_010V137_20180207_075124.dfr	self-ext	self-ext	self-ext	(111.495+110.149j)	-
TEX_010V137_20180211_033100.dfr	self-ext	self-ext	self-ext	(252.664+146.686j)	-
TEX_010V137_20180211_054803.dfr	self-ext	self-ext	self-ext	(88.594+87.12j)	-
VAAR_2.12_20180320_120125.dfr	self-ext	self-ext	self-ext	(31.226+56.8j)	-
WV_7_20180819_020422.dfr	self-ext	self-ext	self-ext	(109.77+159.18j)	-
WV_7_20180819_020430.dfr	self-ext	self-ext	self-ext	(200.554+256.272j)	-
WWF_010V113_20180421_015242.dfr	self-ext	self-ext	self-ext	(50+50j)	-
WWF_010V113_20180601_180825.dfr	self-ext	self-ext	self-ext	(50+50j)	-
WWF_010V113_20180712_102521.dfr	self-ext	self-ext	self-ext	(50+50j)	-
WWF_010V113_20180728_084433.dfr	self-ext	self-ext	self-ext	(50+50j)	-
WWF_010V113_20180816_062251.dfr	self-ext	self-ext	self-ext	(50+50j)	-
WWF_010V113_20180823_143627.dfr	self-ext	self-ext	self-ext	(50+50j)	-
WWF_010V113_20180921_205558.dfr	self-ext	self-ext	self-ext	(50+50j)	-
WWGB_2.55_20180914_122430.dfr	self-ext	self-ext	self-ext	(85.243+82.381j)	-
WWGB_2.75_20211201_045224.dfr	self-ext	self-ext	unstable	(8.879+5.897j)	-
WWGB_2.77_20180321_131946.dfr	self-ext	self-ext	self-ext	(49.245+113.862j)	-
ZBM1_2.15_20190130_131019.dfr	self-ext	self-ext	self-ext	(32.285+32.843j)	-

HBY20_106_20180101_190026.dfr	two-ph	single-ph	two-ph	-	(0.246+0.983j)
SCH1_139_20180108_174247.dfr	two-ph	single-ph	two-ph	(2.591+1.976j)	(0.328+0.173j)
VAAR_2.12_20180320_120118.dfr	self-ext	single-ph	single-ph	(4.681+3.098j)	(5.45+3.878j)
VAAR_2.12_20180320_120121.dfr	self-ext	single-ph	self-ext	(50+50j)	-
VAAR_2.12_20180328_190957.dfr	self-ext	single-ph	single-ph	(50+50j)	(4.867+3.054j)
VAAR_2.12_20180328_190959.dfr	self-ext	single-ph	self-ext	(50+50j)	-
VAAR_2.12_20180328_191003.dfr	self-ext	single-ph	self-ext	(50+50j)	-
VAAR_2.12_20180328_191012.dfr	self-ext	single-ph	single-ph	(4.461+2.58j)	(4.461+2.58j)
DNT_tr5_20210429_123938.dfr	single-ph	unstable	single-ph	(4.63+2.793j)	(5.644+2.869j)
HBY_1.09_20191211_201702.dfr	single-ph	unknown	single-ph	-	(27.479+12.286j)
LEIN_V110_20211115_133704.dfr	single-ph	unstable	single-ph	(1.64+0.88j)	(1.64+0.88j)
LMR_1.09_20210414_111304.dfr	single-ph	unstable	single-ph	(20.304+10.153j)	(20.304+10.153j)
OHK_1.10_20180602_200508.dfr	single-ph	unknown	single-ph	-	(28.44+17.356j)
OHK_1.10_20180622_130824.dfr	single-ph	unknown	single-ph	-	(31.834+14.652j)
ZUI2_KA01_20210101_102033.dfr	single-ph	unstable	single-ph	(6.769+4.821j)	(6.769+4.821j)
ALPC2_V0206_20210130_212604.dfr	unstable	unstable	unstable	(4.486+3.972j)	-
CON2_10_312_20211109_084204.dfr	unstable	unstable	unstable	(174.665+77.102j)	-